

Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Sistemes
Treball de final de carrera

Adaptació d'un portal Web a diferents famílies de dispositius mòbils

Autor: Josep Maria Brunetti Fernández
Directora de la EPS: Montserrat Sendín Veloso
Directora externa: Silvia Penas Casals
Juliol de 2007

Pròleg

Aquest treball neix a principis del 2007. Andreu Ibáñez, gerent i director comercial d'Internet Web Serveis (IWS) em va proposar fer el Treball Final de Carrera amb ells tractant un tema concret: adaptar un portal Web a dispositius mòbils utilitzant una nova tecnologia.

Des del primer moment em va semblar una idea interessant, no només pel tema, totalment desconegut per mi, sinó pel fet de poder estar en una empresa durant aquest temps. Les primeres setmanes em van servir per recopilar informació de la plataforma MyMobileWeb i documents sobre el desenvolupament d'aplicacions per a dispositius mòbils. Tot seguit vaig començar a utilitzar la plataforma, practicant i seguint alguns exemples. Al mateix temps, anava redactant aquest document. Finalment, els 2 últims mesos els he dedicat a posar en pràctica aquests coneixements, adaptant el portal de la Fira de Tàrraga a diferents famílies de dispositius mòbils.

Poder realitzar aquest treball en una empresa m'ha permès mantenir un ritme constant de treball i aprendre, no solament en temes relacionats amb el desenvolupament per a mòbils, sinó també en molts d'altres camps de la informàtica. Però sobretot, he pogut conèixer molta gent que de ben segur no oblidaré.

No voldria deixar d'agrair a totes aquelles persones que han fet possible la realització d'aquest treball i que d'alguna forma hi han contribuït:

En primer lloc, a l'Andreu Ibáñez, per oferir-me la possibilitat de realitzar el treball a IWS, animar-me i confiar en mi. A la Silvia Penas, per fer de codirectora externa del treball, llegir-se aquest document i donar la seva opinió. Al Pol Ferre en especial: moltíssimes gràcies per dedicar-me part del teu temps, resoldre'm tants i tants dubtes i ajudar-me en tots els aspectes tècnics. I a tota la gent de IWS, gràcies per aguantar-me cada dia.

També agraeixo a la Montserrat Sendín, codirectora del treball dins de l'Escola Politècnica Superior, el seu interès des d'un primer moment pel tema del treball. Gràcies també per guiar-me en tot el referent a la realització de la memòria i la presentació.

Gràcies a tots els que han llegit la memòria i m'han permès corregir-ne les errades.

Gràcies als meus companys, amics i família, per donar-me suport i animar-me, no només durant la realització d'aquest treball sinó per tots aquests anys.

I a tots aquells que d'una manera o altra m'han ajudat, tot i que el seu nom no aparegui explícitament en aquestes línies, gràcies.

Índex

1	Introducció	6
1.1	El problema de la mobilitat	6
1.1.1	El context de l'usuari mòbil	7
1.1.2	Heterogeneïtat de dispositius	7
1.1.3	Criteris fonamentals de disseny	7
1.2	Objectius	8
1.3	Estructura del document	8
2	Estat actual de la tecnologia i terminals mòbils	10
2.1	Domini .mobi	10
2.2	Altres iniciatives mòbils	11
2.3	Terminals mòbils	12
2.4	Tecnologies	15
2.4.1	WAP	15
2.4.2	I-MODE	15
2.4.3	GSM - 2G	16
2.4.4	GPRS - 2.5G	16
2.4.5	UMTS - 3G	16
2.4.6	4G	16
2.4.7	Wi-Fi	17
2.4.8	WiMAX	17
3	Descripció de la tecnologia utilitzada	18
3.1	Llenguatges utilitzats	18
3.1.1	XML	18
3.1.2	CSS	19
3.1.3	JAVA	19
3.1.4	JavaScript	19
3.2	Entorn d'execució	20
3.2.1	Apache	20
3.2.2	Tomcat	20
3.2.3	M-CMS	21
3.3	Eines de desenvolupament	21
3.3.1	Eclipse	21
3.3.2	Plugin MyMobileWeb per eclipse	22
3.3.3	Hibernate	22
3.3.4	mDevInf	23

4	MyMobileWeb	24
4.1	Introducció	24
4.2	Característiques principals	25
4.3	Conceptes prèvis	25
4.4	Arquitectura	27
4.4.1	Arquitectura multicanal	27
4.4.2	Arquitectura de desenvolupament	29
4.4.3	Arquitectura d'execució On-Line	29
4.5	La capa de Presentació	30
4.5.1	Definició d'OPs i Presentacions	30
4.5.2	Estructura de les Presentacions	31
4.5.3	Definició d'estils	32
4.5.4	Controls Visuals	32
4.5.5	<i>Binding</i> automàtic de dades	33
	<i>Binding</i> del Terminal al Context	34
	<i>Binding</i> del Context al Terminal	34
4.5.6	JSTL	35
4.5.7	Desplegament de presentacions	35
4.6	Definició i implementació de OAs	36
4.7	Components	36
4.7.1	Gestor de Dispositius	38
	Funcionalitat	39
	Tecnologia base del component: WURFL	39
	Solució adoptada	40
	Procés de reconeixement de dispositius	40
	Arquitectura del component	41
4.7.2	Gestor de Validacions	42
4.7.3	Gestor de Look & Feel i Propietats	42
4.7.4	Gestor de Context	43
4.7.5	Gestor de Missatges i Literals	44
4.7.6	Motor de Flux	45
4.7.7	Gestor de recursos	46
4.7.8	Transcodificador d'imatges	47
4.8	Configuració de l'aplicació en el servidor	47
4.9	Procés de desenvolupament	48
5	Cas pràctic: Fira de Tàrraga	50
5.1	Descripció del sistema existent	50
5.2	Portal Web mòbil de la Fira de Tàrraga	51
5.3	Desenvolupament de l'aplicació	52
5.3.1	Casos d'ús	52
5.3.2	Divisió de la funcionalitat en OPs	53
5.3.3	Descripció de les presentacions	54
5.3.4	Descripció dels estils	55
5.3.5	Descripció de les OAs	55
5.3.6	Descripció de les classes del control de flux	55
5.3.7	Estructura de directoris de l'aplicació	56

6	Proves de validació de l'aplicació	57
6.1	Validador ready.mobi	57
6.2	Proves amb emuladors	58
6.2.1	Nokia N70	58
6.2.2	Sony Ericsson k750i	59
6.2.3	Openwave V7 Simulator	59
6.2.4	Motorola v3i	60
6.2.5	Sharp GX-10	60
7	Conclusions i possibles millores	61
7.1	Conclusions	61
7.2	Treball futur i possible millores	63
8	Annexes	64
8.1	Annex 1 - Exemple d'una presentació en .xml	64
8.2	Annex 2 - Exemple d'una presentació amb tags JSTL	65
8.3	Annex 3 - Fitxer de configuració MyMobileWeb.Global.xml	66
8.4	Annex 4 - Exemple de full d'estil CSS	67
8.5	Annex 5 - Exemple d'una OA	70

Índex de figures

2.1	Dominis .mobi.	10
2.2	Valladolid Movil.	11
2.3	Andalucía Movil.	11
2.4	Telèfons mòbils a Espanya.	12
2.5	Previsió de vendes de mòbils.	12
2.6	Famílies de dispositius mòbils.	13
2.7	Tecnologies dels navegadors mòbils.	13
2.8	Serveis utilitzats en telèfons mòbils.	14
2.9	Gràfica de velocitat i mobilitat de WiMAX.	17
3.1	Arquitectura del servidor.	20
3.2	mDevInf - Mobile Device Information.	23
4.1	Arquitectura multicanal.	28
4.2	Arquitectura d'execució On-Line.	30
4.3	Estructura de directoris de les OPs.	31
4.4	Estructura de directoris dels fulls d'estil.	32
4.5	Arquitectura de Desenvolupament - Desplegament de presentacions.	36
4.6	Arquitectura d'execució On-Line.	37
4.7	Arquitectura del Gestor de Dispositius.	41
4.8	Estructura de directoris dels literals.	44
4.9	Esdeveniments de l'aplicació.	45
4.10	Estructura de directoris dels recursos.	47
5.1	Diagrama de casos d'ús.	52
6.1	Validador ready.mobi.	57
6.2	Proves amb Emulador Nokia N70 - 1.	58
6.3	Proves amb Emulador Nokia N70 - 2.	58
6.4	Proves amb Emulador Sony Ericsson k750i - 1.	59
6.5	Proves amb Emulador Sony Ericsson k750i - 2.	59
6.6	Proves amb Emulador Openwave V7 Simulator - 1.	59
6.7	Proves amb Emulador Motorola v3i - 1.	60
6.8	Proves amb Emulador Sharp GX-10 - 1.	60

Capítol 1

Introducció

1.1 El problema de la mobilitat

Quan es parla de mobilitat s'està fent referència a que l'usuari pugui accedir a certa informació des de qualsevol lloc independentment del tipus de dispositiu que utilitzi.

Actualment, existeix una demanda cada cop major de la disponibilitat incondicional de la Web. Tot i que en els últims anys el mercat dels dispositius mòbils ha anat creixent i s'ofereixen dispositius que permeten realitzar operacions pròpies d'un equip de sobretaula, existeixen limitacions a l'hora d'accedir a certs serveis utilitzant aquests dispositius. En la major part d'ocasions, el resultat és una experiència poc satisfactòria en trobar nombrosos problemes per accedir a la Web des de dispositius mòbils.

Amb l'aparició d'un nombre cada cop major de dispositius diferents en mides i formes, amb característiques tan diverses com càmeres, reconeixement de veu, pantalles tàctils, sistemes de localització... la Web pot assolir un major nombre d'usuaris. La Web mòbil pot ser accessible en llocs on el cable no hi pot arribar. Això comporta noves oportunitats de negoci, oci i molts altres serveis que ens permeten incrementar la nostra capacitat de moviment.

Però per a poder fer ús d'aquesta Web mòbil sense problemes és necessària una adequació dels continguts als dispositius en funció de les seves capacitats. A més a més, cal tenir en compte les diferències entre usuaris mòbils i usuaris fixes: el tipus de contingut que sol·liciten, les característiques dels dispositius (pantalles petites, colors, etc.) i el context en el qual es rep el contingut. Comprendre aquest context és clau per adaptar una Web a dispositius mòbils. L'accés a Internet utilitzant un dispositiu mòbil està influenciat per les circumstàncies i condicions que ens envolten. Res a veure amb l'experiència amb un ordinador on el tamany de la pantalla és pràcticament lliure, l'accés és ràpid i les dades s'introdueixen amb facilitat amb el teclat i el ratolí.

L'accés a la informació des de qualsevol lloc, en qualsevol moment i independentment del dispositiu utilitzat es pot assolir amb aplicacions que s'adaptin dinàmicament a les necessitats de l'usuari, les capacitats del dispositiu i les condicions de l'entorn.

Dotar a les aplicacions i Webs de mobilitat permetrà als usuaris utilitzar diferents dispositius per accedir a la mateixa informació des de qualsevol lloc.

1.1.1 El context de l'usuari mòbil

L'entorn en que l'usuari utilitza un telèfon mòbil o una PDA i les característiques del propi dispositiu són dos elements clau a tenir en compte en el disseny d'aplicacions o Webs per a dispositius mòbils. En el cas d'un dispositiu mòbil, l'entorn es canviant, dinàmic. L'usuari pot estar distret o tenir pressa. Per tant, l'estructura de navegació ha de ser simple i cal evitar els passos innecessaris.

Hi ha alguns aspectes que actualment cal tenir en compte:

- Els usuaris saben que paguen per temps o tràfic.
- En el cas dels telèfons mòbils, la majoria estan optimitzats per a veu, no per a dades.
- La major part dels usuaris es connecten per a realitzar tasques concretes, no per a navegar lliurement.

1.1.2 Heterogeneïtat de dispositius

Els dispositius mòbils tenen unes característiques físiques que afecten negativament a la navegació o a la usabilitat d'una aplicació. El problema més greu és la heterogeneïtat dels dispositius i les seves limitacions. Entre d'altres, aquests són alguns dels factors que dificulten aquesta adaptació:

- Les mides de la pantalla: nombre de línies i caràcters per línia.
- Alguns dispositius poden mostrar imatges en color, altres no.
- Els diferents llenguatges de marcat utilitzats.
- L'ample de banda i inestabilitat de la connexió.

1.1.3 Criteris fonamentals de disseny

1. **Principi KISS:** *Keep It Simple, Stupid*. La traducció literal seria "Mantenir-ho simple, estúpid". En el disseny d'una Web mòbil és necessari fer les coses senzilles i intuïtives. L'usuari ha de ser capaç d'utilitzar el servei sense complicacions. L'estructura de navegació ha de ser senzilla (pocs passos i clars).
2. **Menys és més:** no s'ha de donar informació pel sol fet de tenir-la. Només cal mostrar la informació rellevant i que l'usuari ha sol·licitat.
3. **L'usuari és mòbil:** partint de la base de que gran part dels usuaris es connecten al servei amb objectius concrets i que paguen per un temps o Bytes, és important no alterar el seu camí principal d'acció amb publicitat, continguts no importants o passos innecessaris.
4. **Dificultat de l'escriptura:** escriure resulta difícil en un dispositiu mòbil. Sempre és preferible la selecció d'opcions a l'escriptura. Si l'escriptura és inevitable, aquesta s'ha de reduir el màxim possible.

1.2 Objectius

L'objectiu principal d'aquest treball és adaptar un portal Web ja existent a diferents famílies de dispositius mòbils. Per fer aquesta adaptació s'utilitza la plataforma MyMobileWeb, basada en tecnologies i estàndards oberts.

El portal escollit ha estat el de la Fira de Tàrraga (www.firatarrega.com) i la nova versió adaptada i dissenyada específicament per a dispositius mòbils estarà disponible a mobi.firatarrega.com.

Podem identificar altres objectius com:

- Promocionar la Fira de Tàrraga.
- Mantenir un aspecte homogeni amb el portal Web ja existent, intentant mantenir la mateixa imatge, colors...
- L'aplicació mòbil ha de funcionar com a part integrada del sistema actual. La informació a mostrar s'ha d'obtenir de la base de dades ja existent i ha de ser administrable amb un gestor de continguts.
- Ús del programari lliure per a la realització de totes aquestes tasques.

1.3 Estructura del document

Aquest document està dividit en set capítols, seguits d'annexes que amplien la informació i finalment la bibliografia.

El primer capítol consisteix en una introducció al problema de la mobilitat. S'exposen dos factors a tenir en compte de cara al desenvolupament com són el context de l'usuari mòbil i la diversitat de dispositius. També s'expliquen els criteris fonamentals del disseny per a dispositius mòbils.

En el segon capítol es tracta l'estat actual de les tecnologies dels dispositius mòbils i futures tendències. Es mostren gràfics i estadístiques sobre els telèfons mòbils. També s'expliquen els nous dominis dissenyats per a dispositius mòbils (.mobi) i altres iniciatives en mobilitat.

El tercer capítol correspon a les tecnologies i llenguatges emprats en la realització d'aquest treball, l'entorn d'execució de l'aplicació així com també les principals eines de desenvolupament utilitzades.

El quart capítol és el més tècnic i alhora el més important de tots. En ell es profunditza en la plataforma MyMobileWeb: les seves principals característiques, la seva arquitectura, els seus components, etc.

El cinquè capítol correspon a un cas d'estudi en concret: la Fira de Tàrraga. Es detallen els passos principals que s'han seguit per adaptar el portal de la Fira de Tàrraga a les diferents famílies de dispositius mòbils, utilitzant la plataforma MyMobileWeb.

CAPÍTOL 1. INTRODUCCIÓ

En el sisè capítol es mostren les proves de validació del sistema. Aquestes s'han realitzat utilitzant emuladors de telèfons mòbils, PDAs, i dispositius reals.

El setè i últim capítol presenta les conclusions extretes d'aquest treball a més d'un possible treball futur que permeti introduir alguna millora.

Finalment, s'adjunten annexes que amplien la informació d'algun dels capítols anteriors i també la bibliografia.

Capítol 2

Estat actual de la tecnologia i terminals mòbils

2.1 Domini .mobi

L'extensió .mobi és el primer i únic domini de nivell superior aprovat per la ICANN (Internet Corporation for Assigned Names and Numbers) dedicat als usuaris que accedeixen a Internet des de dispositius mòbils. Es va aprovar al novembre de 2005 i es va afegir al DNS global. El domini .mobi està recolzat i patrocinat per 13 grans companyies de mòbils i Internet: Ericsson, Associació GSM, Google, Hutchison, Microsoft, Nokia, Orascom Telecom, Samsung Electronics, Syniverse, Telefónica Móviles, TIM (Telecom Italia Mobile) , T-Mobile i Vodafone.

Existeix un gran nombre de persones que tenen com a principal punt d'accés a Internet el telèfon mòbil. Amb els dominis .mobi, cadascun d'aquests usuaris pot estar segur d'estar visitant un lloc dissenyat específicament per a dispositius mòbils.

Un domini .mobi és tan fàcil de recordar com qualsevol altre domini .com, .es, etc. Al ser un domini d'alt nivell, no és necessari introduir el prefix "www" i es simplifica l'escriptura en el teclat del dispositiu mòbil.

La figura 2.1 mostra els requisits que ha de complir un domini .mobi.

Desenvolupament de dominis .mobi. Punts clau i requisits
<ul style="list-style-type: none">- Perfil mòbil XHTML.- Accés sense www.- No poden contenir frames, pop-ups ni poden actualitzar-se automàticament.- Navegació senzilla i gràfics apropiats pel dispositiu.

Figura 2.1: Dominis .mobi.

DotMobi, la companyia darrera d'aquest nou domini, ha anunciat que s'han registrat més de 500.000 dominis .mobi en 104 països des del seu llançament comercial.

2.2 Altres iniciatives mòbils

En l'actualitat són varies les ciutats i empreses que s'han decidit a tenir un portal Web accessible des de dispositius mòbils. Deixant de banda les grans empreses, buscadors o multinacionals, aquestes són algunes de les iniciatives mòbils que s'han trobat per la xarxa:

- **EsMadrid movil:** <http://movil.esmadrid.com>
- **Londres:** <http://londonengland.mobi>
- **Manhattan:** <http://manhattan.mobi>

La major part de serveis que ofereixen aquests portals estan relacionats amb el turisme (allotjament, transport, informació turística...), notícies d'actualitat, agenda cultural i previsió meteorològica.

També s'han trobat dos portals Web que han utilitzat la plataforma MyMobileWeb com a eina principal de desenvolupament:

- **Valladolid Movil:** <http://195.235.93.132:8080/VLDMovil/>

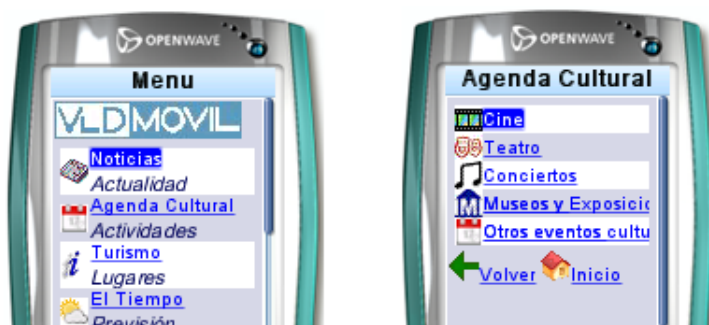


Figura 2.2: Valladolid Movil.

- **Andalucía Movil:** <http://andalucia.mobi>



Figura 2.3: Andalucía Movil.

2.3 Terminals mòbils

Actualment, hi ha més gent que té accés a un telèfon mòbil que a un ordinador de sobretaula. A més a més, per cada ordinador venut, es venen quatre telèfons mòbils. I a Espanya, hi ha més telèfons mòbils que habitants.



Figura 2.4: Telèfons mòbils a Espanya.

Partint d'aquesta base i sabent que la majoria d'usuaris es canvien de mòbil cada un o dos anys, és necessari que la Web sigui accessible des de tots aquests dispositius.

A la figura 2.5 es pot observar la previsió de vendes de telèfons mòbils en els propers anys. Es preveu que al 2009 el nombre d'usuaris de telèfons mòbils arribi als 3000 milions.

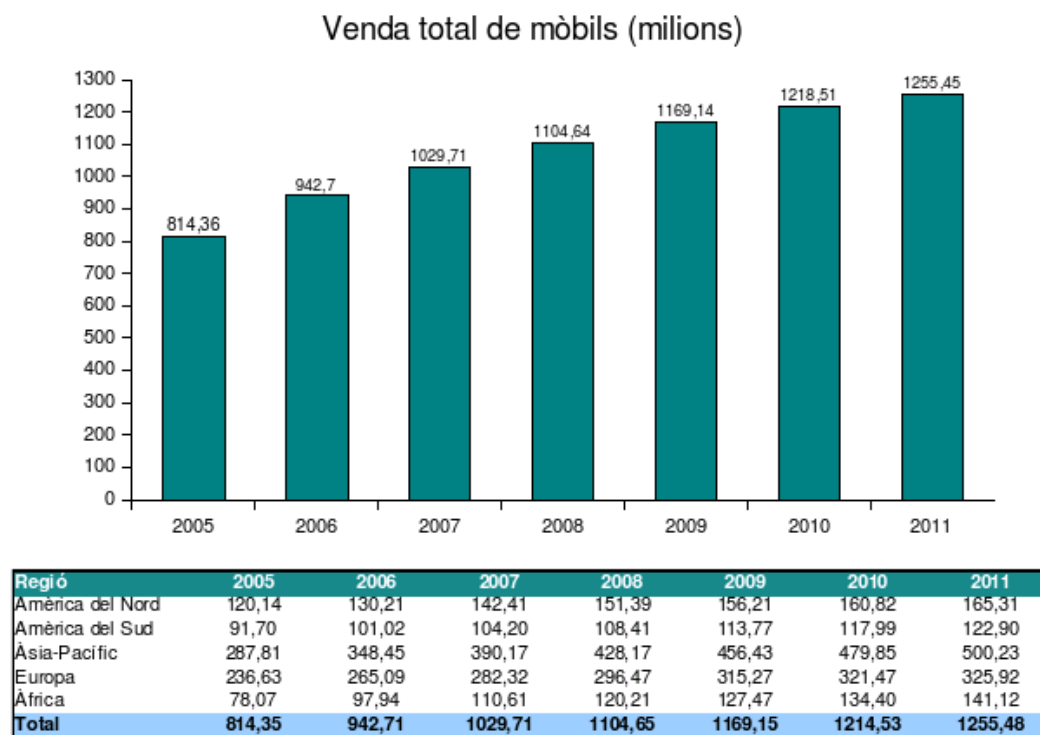


Figura 2.5: Previsió de vendes de mòbils.

Pel que fa a les diferents famílies de dispositius mòbils els podem dividir en:

1. Telèfons mòbils: són la família de dispositius més comuna. Estan adreçats als consumidors en general.

2. Smart Phones: comparteixen les mateixes característiques d'un telèfon mòbil amb la diferència que tenen la capacitat d'executar aplicacions i utilitzen un sistema operatiu més avançat.
3. PDAs: tenen molt en comú amb els Smart Phones però la seva funcionalitat està orientada més aviat cap a l'organització de tasques i no pas a les comunicacions de veu.

Com es pot observar a la figura 2.6, els telèfons mòbils lideren aquest mercat amb un ample marge respecte a les PDAs i els Smart Phones.

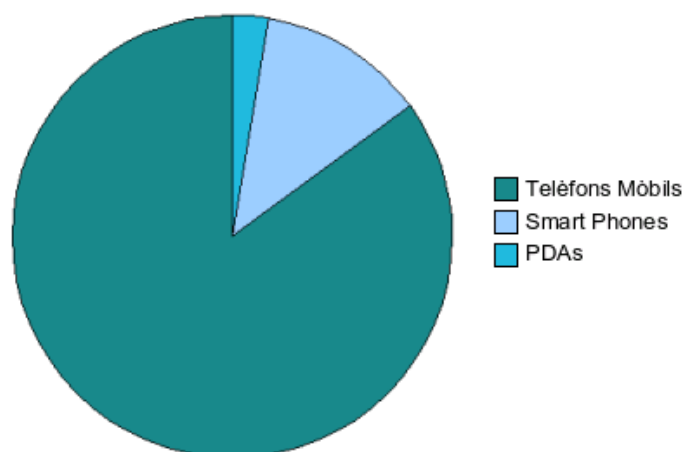


Figura 2.6: Famílies de dispositius mòbils.

Respecte als navegadors utilitzats en dispositius mòbils, la tecnologia més habitual és WAP 2.x/XHTML, tot i que encara hi ha molts navegadors que utilitzen WAP 1.x. La figura 2.7 correspon a un gràfic que compara l'evolució d'aquestes tecnologies amb la tecnologia Wireless dels ordinadors portàtils.

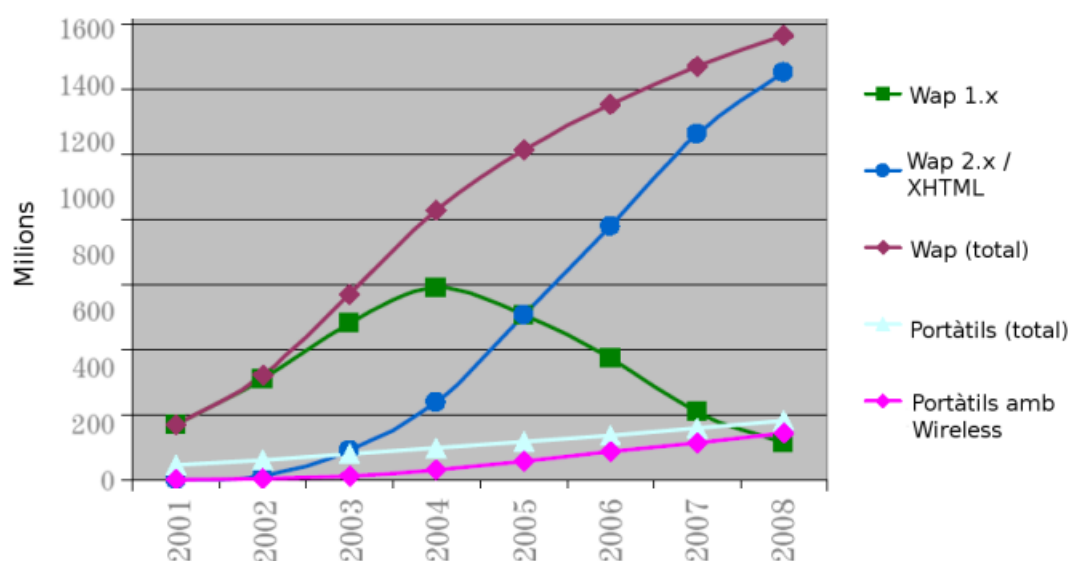


Figura 2.7: Tecnologies dels navegadors mòbils.

La figura 2.8 mostra els diferents serveis utilitzats en telèfons mòbils. El servei més utilitzat és, sense tenir en compte les trucades, l'enviament de missatges SMS. L'accés a Internet des de telèfons mòbils fins ara no ha estat un servei gaire utilitzat degut al seu cost i baixa velocitat. No obstant, l'aparició de noves tecnologies com Wi-Fi per a dispositius mòbils, permetrà un augment de la velocitat i a la vegada una reducció del cost d'aquest servei.

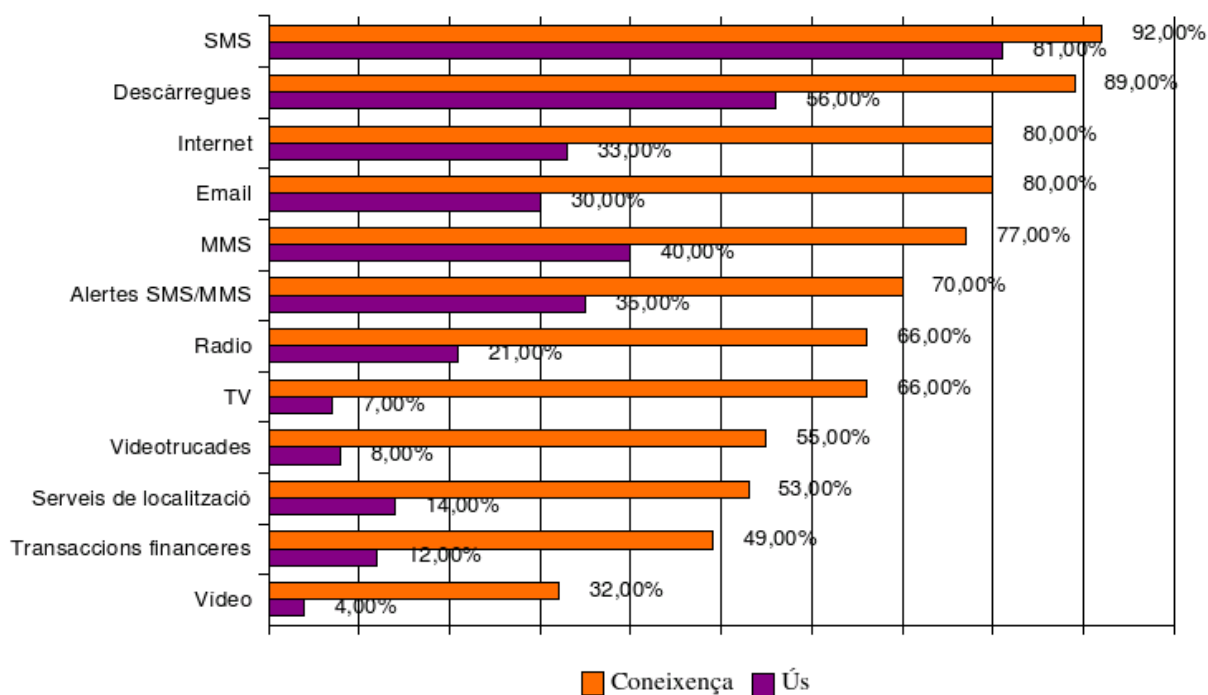


Figura 2.8: Serveis utilitzats en telèfons mòbils.

2.4 Technologies

2.4.1 WAP

La tecnologia WAP (Wireless Application Protocol) és un estàndard obert internacional per a aplicacions que utilitzen comunicacions inal·làmbriques, com l'accés a serveis d'Internet des d'un telèfon mòbil. L'organisme que es va encarregar de desenvolupar l'estàndard WAP va ser el WAP Forum, fundat per quatre empreses del sector de les comunicacions mòbils: Sony-Ericsson, Nokia, Motorola i Openwave. Actualment WAP Forum forma part de la OMA (Open Mobile Alliance), que s'encarrega de la definició de diverses normes relacionades amb les comunicacions mòbils.

Es tracta d'una especificació d'un entorn d'aplicació i d'un conjunt de protocols de comunicacions per a normalitzar el mode en que els dispositius inal·làmbrics accedeixen als diferents serveis. Wap defineix un entorn d'aplicació i una pila de protocols per aplicacions i serveis accessibles a través de terminals mòbils. Consisteix en un conjunt d'especificacions que permeten als desenvolupadors dissenyar aplicacions d'interconnexió per terminals mòbils, habitualment telèfons.

La primera versió de WAP, del 1999, defineix WML (Wireless Markup Language) com a llenguatge de presentació de continguts. Aquesta versió ha estat objecte de fortes crítiques per diferents motius: la pobresa del suport gràfic (imatges en monocrom), les diferències entre les implementacions de WAP en els terminals de diferents fabricants, i problemes de seguretat.

La nova versió WAP 2.0 està present en els telèfons mòbils a partir del 2004. Aquesta versió utilitza XHTML-MP (perfil mòbil) com a llenguatge de presentació de continguts i millora el suport dels gràfics (inclou color). Pel que fa als protocols utilitzats, en la capa de transport s'utilitza TCP i en la d'aplicació HTTP. Així doncs, WAP 2.0 ha adoptat els protocols d'Internet millorant les presentacions d'aquests sobre xarxes de comunicacions mòbils. Els mecanismes de seguretat són comptables amb els d'Internet i els problemes de seguretat de WAP 1 estan resolts.

2.4.2 I-MODE

I-mode és un conjunt de tecnologies i protocols dissenyats per a poder navegar a través de pàgines Web dissenyades específicament per dispositius mòbils. Va ser desenvolupat al 1999 per empreses de telefonia japonesa i inclou suport d'imatges i mòbils amb color.

Aquesta tecnologia competeix amb WAP i altres tipus de tecnologies de navegació per dispositius mòbils. Mentre WAP 1.0 utilitza WML com a llenguatge de presentació de continguts, I-Mode utilitza C-HTML, basat en HTML.

I-Mode és molt popular al Japó amb més de 46 milions de consumidors. A Espanya, Telefónica Movistar és l'operadora que ofereix aquesta tecnologia.

2.4.3 GSM - 2G

GSM (Global System for Mobile communications) és un estàndard mundial per a telèfons mòbils digitals. És un estàndard obert, no propietari i evolutiu. Al 2001, el 70% dels telèfons mòbils del món utilitzaven GSM.

La tecnologia GSM es pot dedicar tan a veu com a dades. Una connexió de dades permet a l'usuari utilitzar el telèfon mòbil com un mòdem de 9600 bps.

2.4.4 GPRS - 2.5G

GPRS (General Packet Radio Service) és una tecnologia digital de telefonia mòbil i es considera la generació 2.5, entre la segona generació (GSM) i la tercera generació (UMTS).

Proporciona altes velocitats de transferència de dades i s'utilitza en les xarxes GSM. GPRS és només una modificació de la forma de transmetre dades en una xarxa GSM, passant de la commutació de circuits de GSM a la commutació de paquets. Aquest canvi permet la compartició dels recursos de ràdio: un usuari GPRS només utilitza la xarxa quan s'envia o es rep un paquet d'informació.

Els avantatges respecte GSM són la velocitat de recepció de dades, que augmenta de 9'6 kbps a 40 kbps, i la tarifació per part de l'operador de telefonia mòbil, que només es produeix per informació transmesa i no per temps de connexió.

2.4.5 UMTS - 3G

UMTS (Universal Mobile Telecommunications System) és una de les tecnologies utilitzades pels mòbils de tercera generació (3G). És el successor de GSM i GPRS. UMTS permet introduir molts més usuaris simultàniament a la xarxa global del sistema i permet incrementar la velocitat fins a 2 Mbps.

La tecnologia HSDPA (High Speed Downlink Packet Access), que permet velocitats de baixada de fins a 14 Mbps, i HSUPA (High Speed Uplink Packet Access), que permet velocitats de pujada de fins a 5.8 Mbps, són l'evolució de 3G i s'anomenen 3.5G i 3.75G respectivament.

2.4.6 4G

4G són les sigles de la quarta generació de tecnologies de telefonia mòbil. 4G no és una tecnologia o estàndard definit, sinó una col·lecció de tecnologies i protocols per a permetre el màxim rendiment amb la xarxa in-làmbrica més barata.

Més que un salt radical, la quarta generació de telefonia mòbil és un nou pas de les telecomunicacions cap a la mobilitat i accessibilitat universals. La quarta generació estarà basada totalment en IP i serà compatible amb Internet i altres tecnologies com Wi-Fi i WiMAX. S'espera que 4G sigui la convergència de les tecnologies al·làmbriques i inal·làmbriques.

2.4.7 Wi-Fi

Wi-Fi (o Wifi) és un conjunt d'estàndards per a xarxes inal·làmbriques basats en les especificacions IEEE 802.11. Es va crear per a ser utilitzat en xarxes inal·làmbriques locals però actualment s'utilitza per accedir a Internet.

La tecnologia Wi-Fi s'ha estès ràpidament en hotels, aeroports, cafeteries, estacions de servei... proveint l'accés a Internet en entorns mòbils. Els nous dispositius mòbils ja incorporen Wi-fi i permeten la connexió a Internet a altes velocitats i amb un cost molt més reduït.

2.4.8 WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) és un estàndard de transmissió inal·làmbrica de dades proporcionant accés en àrees de fins a 50 Km de radi i velocitats de fins a 70 Mbps, utilitzant una tecnologia que no requereix visió directa de les estacions base.

La figura 2.9 mostra una gràfica de la velocitat i mobilitat de la tecnologia WiMAX en comparació amb altres tecnologies.

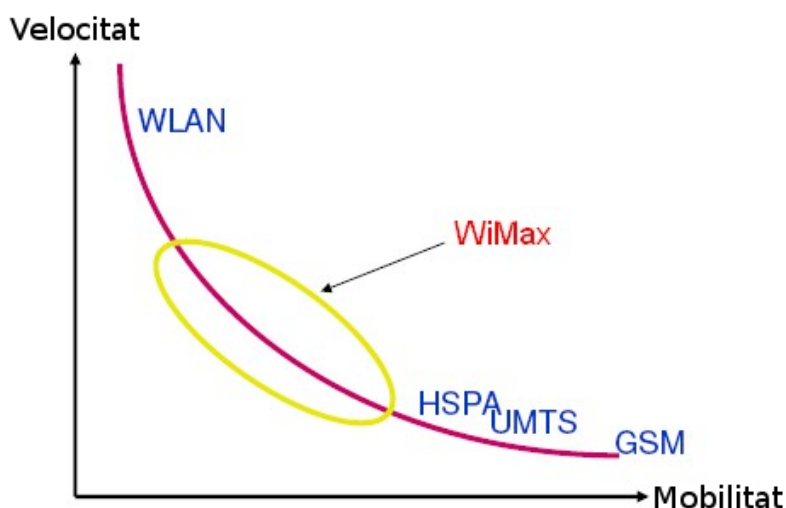


Figura 2.9: Gràfica de velocitat i mobilitat de WiMAX.

La tecnologia WiMAX pot canviar la forma en que es realitzin les connexions inal·làmbriques a Internet a les ciutats. Les companyies de telecomunicacions poden donar cobertura amb poques estacions repartides estratègicament, a un preu molt menor degut a que no es necessari el cablejat ni el seu manteniment.

Capítol 3

Descripció de la tecnologia utilitzada

3.1 Llenguatges utilitzats

En aquesta secció es descriuen els llenguatges de programació utilitzats. La plataforma MyMobileWeb utilitza XML i CSS per a la definició de les presentacions, mentre que el control de flux de l'aplicació es basa en Java. Per al gestor de validacions s'utilitza JavaScript i per a la traducció dels literals fitxers XML.

3.1.1 XML

XML són les sigles de eXtensible Markup Language (llenguatge de marques extensible). XML és un metallenguatge extensible d'etiquetes desenvolupat pel World Wide Web Consortium (W3C). XML prové d'un llenguatge inventat per IBM als anys 70 anomenat GML (General Markup Language).

XML no s'aplica només a Internet sinó que es proposa com un estàndard per l'intercanvi d'informació estructurada entre diferents plataformes. Es pot utilitzar en bases de dades, editors de text, fulls de càlcul... XML és una tecnologia senzilla que es complementa amb d'altres i així augmenta les seves possibilitats.

Característiques principals de XML:

- És una arquitectura oberta i extensible.
- Major consistència i homogeneïtat dels identificadors descriptius d'un document XML en comparació amb els de HTML.
- L'extensibilitat i flexibilitat d'aquest llenguatge permet agrupar una gran varietat d'aplicacions: des de pàgines Web fins a bases de dades.
- Facilitat per exportar les dades a altres formats de publicació.
- Els documents XML han de seguir una estructura estrictament jeràrquica respecte a les etiquetes que delimiten els seus elements.

La tecnologia XML busca donar una solució al problema d'expressar informació estructurada de la manera més abstracta i reutilitzable possible.

3.1.2 CSS

Els CSS (Cascading Style Sheets - Fulls d'Estil en Cascada) són un llenguatge formal que s'utilitza per definir la presentació d'un document estructurat escrit en HTML o XML. El W3C (World Wide Web Consortium) és l'organisme encarregat de l'especificació dels fulls d'estil que serveix com a estàndard.

La idea que es troba darrera dels CSS és separar l'estructura d'un document de la seva presentació. Quan s'utilitza CSS, les etiquetes dels documents HTML o XML no proporcionen informació sobre com ha de ser la presentació, sinó que únicament marquen l'estructura del document. El corresponent full d'estil s'encarrega d'especificar com s'ha de mostrar aquella etiqueta: color, font, alineació del text, mida...

Els principals avantatges d'utilitzar CSS són:

- Control centralitzat de la presentació d'una Web que agilitza qualsevol tipus d'actualització dels continguts.
- Els navegadors permeten als usuaris especificar el seu propi full d'estil. Així s'augmenta l'accessibilitat. Per exemple: persones amb deficiències visuals poden configurar el seu propi full d'estil augmentant el tamany del text o remarcant enllaços.
- Una pàgina pot disposar de diferents fulls d'estil per diferents famílies de dispositius o fins i tot per a que l'usuari pugui escollir.

3.1.3 JAVA

Java és un llenguatge de programació orientat a objectes desenvolupat per Sun Microsystems a principis dels anys 90. Es necessita tenir una màquina virtual instal·lada per a poder executar el codi compilat amb aquest llenguatge.

Aquestes són les principals característiques de JAVA:

- Metodologia de programació orientada a objectes.
- Permet l'execució d'un mateix programa en múltiples sistemes operatius.
- Permet executar codi en sistemes remots de forma segura.
- És fàcil d'utilitzar i utilitza el millor d'altres llenguatges orientats a objectes com C++.

A finals del 2006 Sun va anunciar que lliberaria el codi de Java SE (Standard Edition), Java EE (Enterprise Edition) i Java ME (Micro Edition).

3.1.4 JavaScript

JavaScript és un llenguatge interpretat i no necessita cap compilació. S'utilitza principalment en pàgines Web i la seva sintaxis és semblant a la de Java i C. Al contrari que Java, JavaScript no és un llenguatge orientat a objectes ni disposa d'herència. Més aviat és un llenguatge basat en prototips i les noves classes es generen clonant les classes base (prototips) i extenent la seva funcionalitat.

El principal problema de JavaScript és que el codi s'executa en el client i per tant és visible i pot ser llegit per l'usuari.

Tots els navegadors interpreten el codi JavaScript integrat en les pàgines Web. JavaScript es pot incloure en un document HTML o en qualsevol que s'acabi traduint a HTML en el navegador del client.

3.2 Entorn d'execució

La figura 3.1 mostra l'arquitectura del servidor on s'allotja l'aplicació. (explicar)

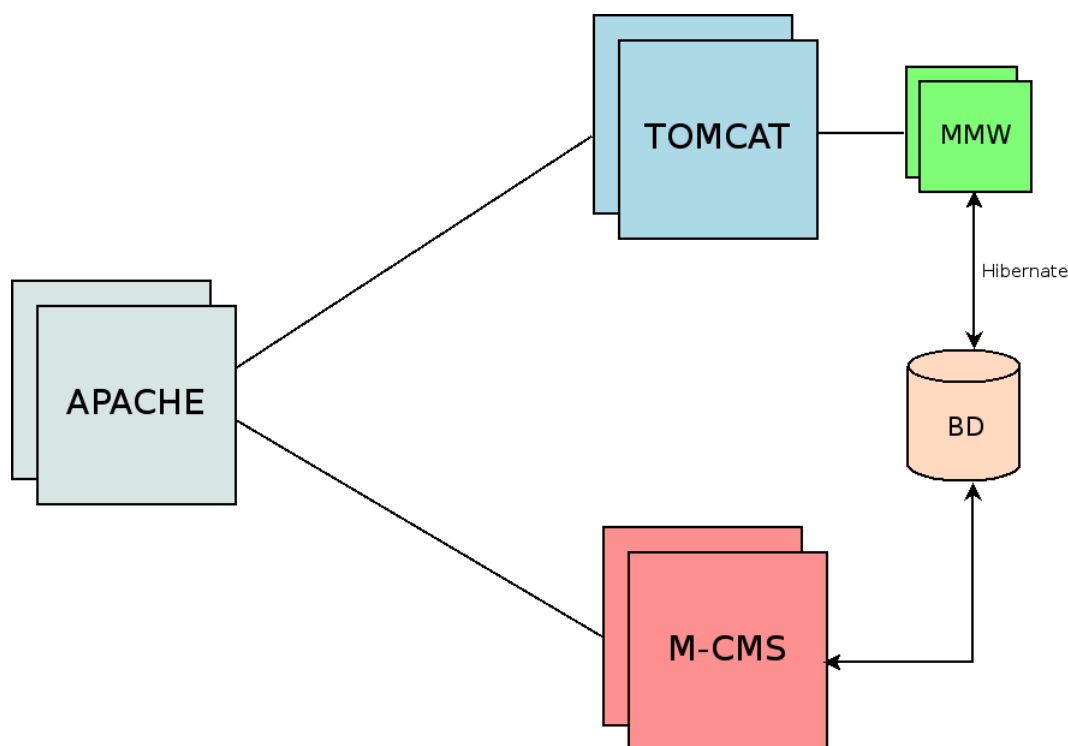


Figura 3.1: Arquitectura del servidor.

3.2.1 Apache

Apache és un servidor HTTP de codi obert multiplataforma. Es pot executar en entorns Unix, Windows, Macintosh i altres. Apache té una alta acceptació a Internet: és el servidor més utilitzat, sent servidor HTTP de més del 70% de Webs del món i continua creixent.

Un servidor Web s'encarrega de romandre a l'espera de peticions HTTP de diferents clients. Els navegadors realitzen una petició al servidor i aquest respon amb el contingut que el client sol·licita.

3.2.2 Tomcat

Tomcat, també anomenat Jakarta Tomcat o Apache Tomcat, és un servidor Web amb suport de servlets i JSP, desenvolupat per Apache Software Foundation. Habitualment, el servidor

d'aplicacions Tomcat s'utilitza combinant-lo amb el servidor Web Apache ja que aquest és més ràpid servint peticions.

Es denomina servidor d'aplicacions a un servidor que executa certes aplicacions. Habitualment es tracta d'un software que proporciona serveis d'aplicació a les computadores client. El servidor d'aplicacions gestiona la major part (o totalitat) de les funcions lògiques i d'accés a les dades de l'aplicació.

Per tal de tenir integrat el servidor d'aplicacions dins d'Eclipse s'ha instal·lat el plugin de Tomcat. D'aquesta manera, per executar les aplicacions locals només cal iniciar Tomcat des d'Eclipse i accedir amb el navegador a <http://localhost:8080> (8080 és el port que utilitza Tomcat per defecte).

3.2.3 M-CMS

Un dels objectius del treball era integrar l'aplicació mòbil amb el sistema existent actualment. Tota la informació que es mostrarà a l'usuari s'ha d'obtenir de la base de dades de la Fira de Tàrraga.

M-CMS (Mobile Content Management System) és el gestor de continguts d'IWS. En aquest cas controla la base de dades de la Fira de Tàrraga i permet crear i administrar els continguts de l'aplicació mòbil amb facilitat. D'aquesta manera la nostra aplicació serà dinàmica i un canvi en els continguts no suposarà un canvi en la programació.

3.3 Eines de desenvolupament

3.3.1 Eclipse

Per la realització d'aquest treball s'ha utilitzant un Entorn Integrat de Desenvolupament (IDE). Un IDE és un entorn de programació empaquetat com una aplicació que proporciona un marc de treball amigable pel programador.

L'IDE escollit pel desenvolupament ha estat Eclipse, dissenyat i desenvolupat per IBM. Actualment, Eclipse té una comunitat d'usuaris que van estenent constantment les àrees d'aplicació cobertes.

Eclipse és una plataforma d'eines universal, multilinguatge i multiplataforma que proporciona un marc de treball o framework per desenvolupar aplicacions i noves eines. L'entorn de treball que presenta eclipse té diferents perspectives personalitzables.

El principal avantatge d'Eclipse en vers altres IDEs és que està basat en mòduls o plugins. Aquest mecanisme de mòduls permet a Eclipse estendre's a altres llenguatges de programació com C/C++, Python, PHP... També permet treballar amb llenguatges pel processament de text com LaTeX, aplicacions en xarxa com Telnet i amb Sistemes de gestió de bases de dades.

Eclipse està escrit majoritàriament en Java (més d'un 90% del codi) i és necessari per a la seva execució que hi hagi un JRE (Java Runtime Environment) instal·lat prèviament en el sistema.

3.3.2 Plugin MyMobileWeb per eclipse

En aquest cas, existeix un Plugin de la plataforma MyMobileWeb per a Eclipse que facilita el desenvolupament d'aplicacions mòbils.

Per a utilitzar aquest plugin es requereix tenir instal·lat:

- Apache Tomcat: servidor d'aplicacions amb suport de servlets i JSPs.
- Plugin de Tomcat per a Eclipse: plugin que integra el servidor Tomcat amb la plataforma de desenvolupament Eclipse.
- Plugin Colorer: opcionalment podem utilitzar aquest plugin que permet trobar amb facilitat errors de sintaxi en els fitxers XML.
- SDK (Software Development Kit) de MyMobileWeb.

Aquest plugin permet crear amb facilitat noves OPs, presentacions, OAs i fulls d'estil. També ens crea els fitxers de configuració .xml del servidor de forma automàtica amb uns valors per defecte. El plugin permet també generar totes les presentacions, el codi de l'aplicació i extreure'n els literals de forma automàtica.

3.3.3 Hibernate

Hibernate és una eina de programari lliure de mapeig de base de dades desenvolupada en Java. Facilita el mapeig de taules i els seus camps d'un model relacional al model d'objectes (atributs) d'una aplicació. Això es fa mitjançant arxius XML que permeten establir aquestes relacions.

La característica principal d'Hibernate és el mapeig de classes en Java a taules d'una base de dades (i de tipus de dades de Java a tipus de dades de SQL), oferint també la possibilitat de realitzar consultes de dades i facilitats de recuperació. Hibernate genera les sentències SQL i allibera al programador del maneig de les dades que resulten de l'execució d'aquestes sentències.

Aquest és el procés de desenvolupament que s'ha seguit utilitzant hibernate:

1. Crear l'arxiu XML de propietats perquè Hibernate conegui tota la configuració JDBC per accedir a la base de dades.
2. Crear una taula SQL per a guardar els objectes persistents.
3. Crear un JavaBean que representi aquest objecte.
4. Crear un arxiu de mapeig de manera que Hibernate sapigui quins atributs de la classe corresponen a cada camp de la taula SQL.
5. Utilitzar l'API d'Hibernate.

3.3.4 mDevInf

mDevInfo (Mobile Device Information) proporciona una manera fàcil i amigable d'accedir a la informació que conté la base de dades WURFL¹. Permet consultar les característiques de qualsevol dispositiu: la mida de la pantalla, els formats suportats, la tecnologia de marcat que utilitza, etc. Això és útil quan es vol fer un disseny per a una família de dispositius en concret i es necessita conèixer les seves característiques.

mDevInfo és una eina útil no només per a programadors, sinó també per usuaris convencionals ja que els permet conèixer millor els seus dispositius.

La figura 3.2 mostra algunes de les característiques d'un dispositiu mòbil, en concret els llenguatges de marcat que suporta.

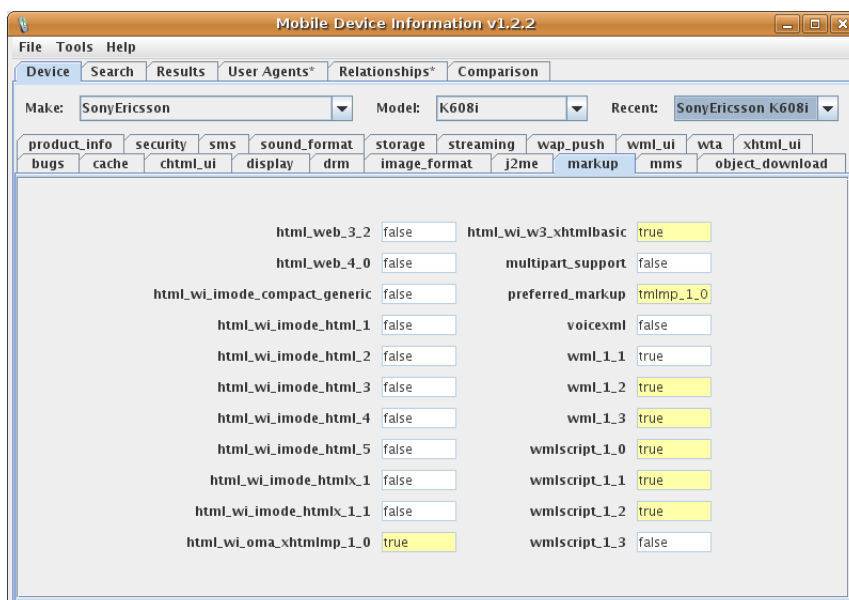


Figura 3.2: mDevInf - Mobile Device Information.

¹És un repositori on s'emmagatzemen les capacitats dels dispositius inal·làmbrics. En el quart capítol s'explica amb més detall.

Capítol 4

MyMobileWeb

4.1 Introducció

MyMobileWeb és una plataforma composta per un conjunt de components i eines que permeten el desenvolupament d'aplicacions mòbils basades en clients lleugers (WAP 1.x, WAP 2.0, i-Mode i navegadors de PDA). Està basat en estàndards, consta de diferents mòduls i ofereix una solució completa al problema de la mobilitat.

MyMobileWeb proporciona els elements necessaris per adaptar les presentacions a les diferents famílies de dispositius mòbils i una plataforma de components basada en una arquitectura Model-Vista-Controlador (MVC) que accelera el desenvolupament de les aplicacions.

El model de canal es recolza en els principis d'Arquitectures Orientades a Serveis (SOA):

- Les aplicacions exporten serveis accessibles a través de diferents canals: portal/canal Web tradicional i portal/canal de mobilitat.
- Els serveis són independents del canal a través del qual s'hi accedeix.
- No es dupliquen els serveis.

Escenaris d'aplicació de MyMobileWeb:

- Sistemes d'Informació i Serveis accessibles des de múltiples dispositius.
- Escenaris on es requereix una interacció tan on-line com off-line
- Aplicacions existents per les quals es vol mobilitzar la seva funcionalitat.

En aquest cas, la plataforma MyMobileWeb s'ha emprat per a realitzar una aplicació totalment on-line. És per això que no s'ha detallat l'arquitectura o els components off-line de la plataforma.

4.2 Característiques principals

- Permet el Desenvolupament Ràpid d'Aplicacions (RAD) i Serveis Multicanal i Multidispositiu en temps i cost rècord.
- Únicament són necessàries habilitats de desenvolupament convencionals: Java, Web i XML.
- Es basa en controls visuals avançats per a mobilitat.
- Arquitectura d'altres prestacions ja que elimina la necessitat de transcodificació¹ per cada petició en execució.
- Permet la integració amb WURFL per la gestió d'informació de les diferents famílies de dispositius.
- Es permet codi JavaScript de validació i gestió de literals.
- Gestió intel·ligent de la paginació per controls visuals, per paràgrafs...
- Independència respecte al servidor Web/aplicacions.
- La plataforma incorpora la capacitat de desplegar aplicacions en un entorn off-line amb una posterior sincronització

4.3 Conceptes prèvis

La plataforma MyMobileWeb utilitza una sèrie de conceptes bàsics que cal conèixer. A continuació es descriuen breument:

- **Expression Language (E.L.):** és el llenguatge de definició d'expressions adoptat per JSTL i l'estàndard JSP 2.0. Permet fer referència de forma simplificada dades presents en el context, seguint una notació puntejada.
- **Operació de Presentació (OP):** una OP és un conjunt de presentacions més el flux que hi ha entre elles, que permeten satisfer un cas d'ús de l'usuari. Per exemple: mostrar els detalls d'un producte seria una OP que permetria a l'usuari veure aquest producte i estaria composta per les diferents presentacions, que implementen la interfície d'usuari, i el flux entre elles.
- **Pseudo-Operació de Presentació:** són OPs que apareixen habitualment en totes les aplicacions i que són gestionades internament per la plataforma MyMobileWeb:
 - Login: gestiona l'entrada a l'aplicació mitjançant un nom d'usuari i una contrasenya (pot no utilitzar-se).
 - Welcome: operació de benvinguda a l'aplicació.
 - Messages: incorpora les presentacions que permeten mostrar diferents missatges a l'usuari (advertències, errors...).
- **Presentació:** és un conjunt de controls visuals agrupats mitjançant un *layout* (disposició). Les presentacions de MyMobileWeb s'especifiquen amb XML, abstraient així les particularitats dels llenguatges de marques dels diferents dispositius.

¹Consisteix en convertir dades i aplicacions escrites en llenguatge Web a formats llegibles en dispositius mòbils.

- **Context:** el context és un “magatzem” de dades jeràrquic. En un moment concret conté les dades que s’estan manejant en l’aplicació. Dins del patró Model-Vista-Controlador el context és el contenidor del Model. Existeixen cinc nivells de context:
 - Aplicació: emmagatzema dades visibles per tots els usuaris d’una aplicació. En aquest nivell del context estaran aquelles dades de l’aplicació que siguin configuracions globals.
 - Sessió: emmagatzema les dades visibles en la sessió d’un usuari. Habitualment es guardaran dades com l’idioma, el perfil de l’usuari, etc. Les dades presents en el context de sessió es mantindran fins que no es finalitzi la sessió de l’usuari.
 - OP: emmagatzema dades associades a l’Operació de Presentació en curs. Quan es canvia d’Operació de Presentació aquestes dades desapareixen.
 - Presentació: emmagatzema dades associades a la Presentació visualitzada per l’usuari en cada moment. Quan es canvia de presentació aquestes dades desapareixen.
 - Request: emmagatzema dades associades al processament de la petició realitzada per l’usuari. Quan la resposta s’ha enviat al terminal aquestes dades desapareixen.
- **Control Visual:** un control visual és un element d’interacció amb l’usuari. Quan es defineix un control visual se li associen dos elements: un *bind*, que és l’element del context on es disposarà la dada associada al control, i un *optionsbind*, que és l’element del context del que obté totes les seves dades el control.
- **Binding Automàtic:** permet associar de forma automàtica l’element de dades del que un control s’alimenta i viceversa.
- **Operació d’Aplicació (OA):** és el punt d’entrada a la lògica específica del domini de l’aplicació. Agafa les dades del context, crida als serveis o components necessaris i deixa noves dades en el context.
- **Inicialitzador de l’Aplicació:** és la classe encarregada de dur a terme les operacions d’inicialització d’una aplicació en el servidor. S’executa una sola vegada durant la vida del servidor per cada aplicació. Aquesta classe és invocada pel framework MyMobileWeb.
- **Gestor de Flux o de Diàleg:** és el component encarregat de gestionar el diàleg de l’usuari amb l’aplicació. El Control de Flux ha de decidir en funció del control, l’event i les dades del context, de la OP i presentació, quina és la següent acció a executar. Normalment aquesta acció serà executar una OA (per a obtenir dades i deixar-les en el context) i navegar cap a la següent presentació.
- **Gestor de Login:** s’encarrega de gestionar l’entrada de l’usuari a l’aplicació. S’invoca automàticament pel framework MyMobileWeb. La plataforma proporciona gestors de login predefinits que permeten autenticar a l’usuari mitjançant directoris LDAP², autenticació bàsica o altres mètodes.
- **Gestor de Perfil d’usuari:** s’encarrega de calcular el perfil d’usuari. Aquest component és invocat automàticament cada cop que es crea una nova sessió d’usuari.

² *Lightweight Directory Access Protocol*, és un protocol de tipus client-servidor per accedir a un servei de directori. El servei més important que proporciona és el d’un repositori d’informació de comptes amb usuaris i claus.

- **Gestor de Dispositius:** és el component que s'encarrega de reconèixer en cada moment el dispositiu que utilitza l'usuari i les seves capacitats. Es basa en WURFL amb extensions pròpies.
- **Gestor de Propietats i Estils:** és el component que s'encarrega de determinar en cada moment quines són les propietats associades a cada control visual en funció del que ha especificat el programador i en funció de la tecnologia i família del dispositiu pel que es genera la presentació.
- **Validador automàtic:** és el component que implementa la lògica necessària per a que les validacions es realitzin de forma automàtica en funció del que hagi especificat el programador en temps de disseny. Les validacions poden ser locals (en el dispositiu si aquest suporta llenguatge de script) o en servidor (si el dispositiu no suporta llenguatge de script).
- **Gestor de Missatges:** s'encarrega de mostrar missatges als usuaris en diferents idiomes. Aquests missatges són parametrizables.
- **Gestor de Literals:** s'encarrega de mostrar els literals de l'aplicació a l'usuari en l'idioma que li correspongui. A més, és possible la redefinició de literals per a famílies i/o tecnologies concretes. D'aquesta manera és poden retallar certs literals degut a les limitacions de pantalla d'alguns dispositius.
- **Extractor de Literals:** és el component encarregat d'extreure els literals d'una aplicació, permeten la generació dels fitxers de literals multi-idioma. Aquest component permet que el programador desenvolupi les aplicacions en el seu idioma natiu sense necessitat de manejar identificadors de literals.
- **Catàleg de Continguts:** és un document XML que recull la informació sobre els recursos, habitualment imatges, utilitzats per l'aplicació (mides, colors, etc.).
- **Generador de Presentacions:** és l'eina que genera les presentacions especificades pel programador en temps de disseny. Parteix del XML + W-CSS que defineix la presentació i genera pàgines JSP adaptades als dispositius. També genera el codi de script necessari per a realitzar les validacions.

4.4 Arquitectura

4.4.1 Arquitectura multicanal

MyMobileWeb incorpora un DriverHTTP encarregat de rebre peticions HTTP provinents de diferents clients: dispositius mòbils, PCs... Quan el DriverHTTP detecta que la petició prové d'un dispositiu mòbil, delega aquesta petició a l'Adaptador de Canal de Mobilitat. Aquest adaptador s'encarrega de resoldre els aspectes de presentació i adaptació (front-end) per a l'entorn mòbil.

La figura 4.1 mostra un esquema de l'arquitectura de MyMobileWeb.

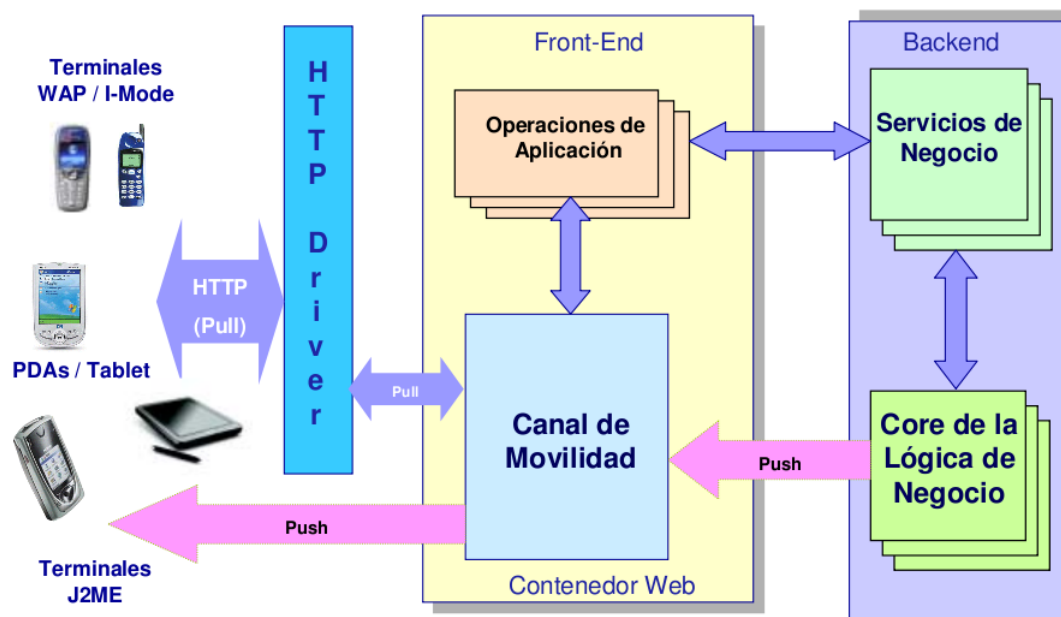


Figura 4.1: Arquitectura multicanal.

- **Driver HTTP:** el Driver HTTP és l'element software que rep les peticions HTTP realitzades pels clients: les de dispositius mòbils, com les d'ordinadors convencionals. Conté la lògica necessària per redirigir la petició a l'adaptador de canal corresponent en funció del client que ha realitzat la petició.
- **Adaptador de Canal de Mobilitat:** la funció de l'Adaptador de Canal de Mobilitat és invocar les operacions de negoci associades a la sol·licitud de l'usuari i adaptar les dades de sortida d'aquestes operacions al format suportat pels terminals.
- **Push Driver:** s'encarrega de donar suport a les comunicacions push. Aquestes consisteixen en notifikacions enviades des del servidor cap a un o varis terminals mòbils. El model de comunicació push (des del servidor cap al client) és el complementari del model de comunicació pull (del client cap al servidor), que és el més tradicional en les aplicacions client-servidor.
- **Operacions d'Aplicacions (OAs):** és el punt d'entrada a la lògica de negoci pel canal de mobilitat. Aquest component és el que s'encarrega d'obtenir noves dades i sol·licitar l'execució d'una funció de negoci.
- **Serveis de Negoci:** són els serveis presents en tota arquitectura SOA. Els serveis de negoci són el punt d'accés a la lògica de negoci a través d'uns mètodes ben definits.
- **Core de la Lògica de Negoci:** són els elements software que contenen dins del *backend* tot el nucli de la lògica específica del domini de negoci. Aquest *core* es invoca pels serveis de negoci.

4.4.2 Arquitectura de desenvolupament

El desenvolupament és realitza com en una aplicació Web J2EE convencional que inclourà, de forma adicional, les llibreries de la plataforma de mobilitat. La capa de presentació es defineix en XML + W-CSS per a posteriorment ser desplegada. La creació de les presentacions en llenguatge conegut per cada dispositiu es realitza de forma automàtica en temps de desenvolupament i no d'execució.

L'Arquitectura de Desenvolupament està orientada a definir Operacions de Presentació (OPs) o Casos d'ús. Cada Cas d'ús estarà format per un conjunt de presentacions. Les presentacions de les OPs poden estar definides a nivell genèric (multidispositiu) o bé redefinir-se per famílies de dispositius a conveniència del desenvolupador. Aquestes presentacions es defineixen mitjançant XML i estan formades per controls visuals.

Un cop es disposa de l'especificació de les presentacions en XML+W-CSS, mitjançant l'Eina de Desplegament de Presentacions s'executa un Generador Automàtic de Codi de:

- De marques + JSP:
 - i-HTML
 - i-XHTML (i-Mode Step 3)
 - XHTML-MP
 - HTML 3.2
 - WML 1.1
- MIDP + J2ME-Polish

Aquest generador automàtic de codi s'encarrega de construir pàgines JSP en els diferents llenguatges que maneja la plataforma. En el temps d'execució, aquestes pàgines JSP juntament amb el seu *runtime* associat, s'encarregaran de realitzar la presentació per a la família de dispositius concreta.

4.4.3 Arquitectura d'execució On-Line

Les peticions HTTP arriben al Canal de Mobilitat provinents dels micronavegadors dels terminals mòbils. Aquestes peticions contenen l'identificador del control visual sobre el que ha actuat l'usuari, l'acció provocada sobre el control visual (onsubmit, onclick...) i les dades (si existeixen) que acaba d'introduir l'usuari.

Un cop el DriverHTTP rep la petició i l'encamina cap a l'Adaptador de Canal de Mobilitat, aquesta petició es rebuda per un Controlador que coordina el procés de reconeixement de dispositiu (un cop per sessió) i la delegació de la sol·licitud a un Motor de Flux d'execució. El Motor de Flux, en funció del control, de l'acció i de les dades del context, decidirà les accions a realitzar. Habitualment, aquestes accions consistiran en l'execució d'Operacions d'Aplicació (OAs) que deixaran noves dades en el context.

Finalment, s'haurà de mostrar una nova presentació a l'usuari. El Motor de Flux sol·licitarà que es mostri a l'usuari la presentació corresponent, que desencadenarà un procés d'elecció de la pàgina JSP (generada en temps de desenvolupament) que contindrà el codi de marcat preferit pel dispositiu. Aquesta pàgina JSP en col·laboració amb un *runtime* realitzarà la presentació al dispositiu de destí.

La figura 4.2 mostra l'arquitectura d'execució del Canal de Mobilitat en el cas de les aplicacions on-line.

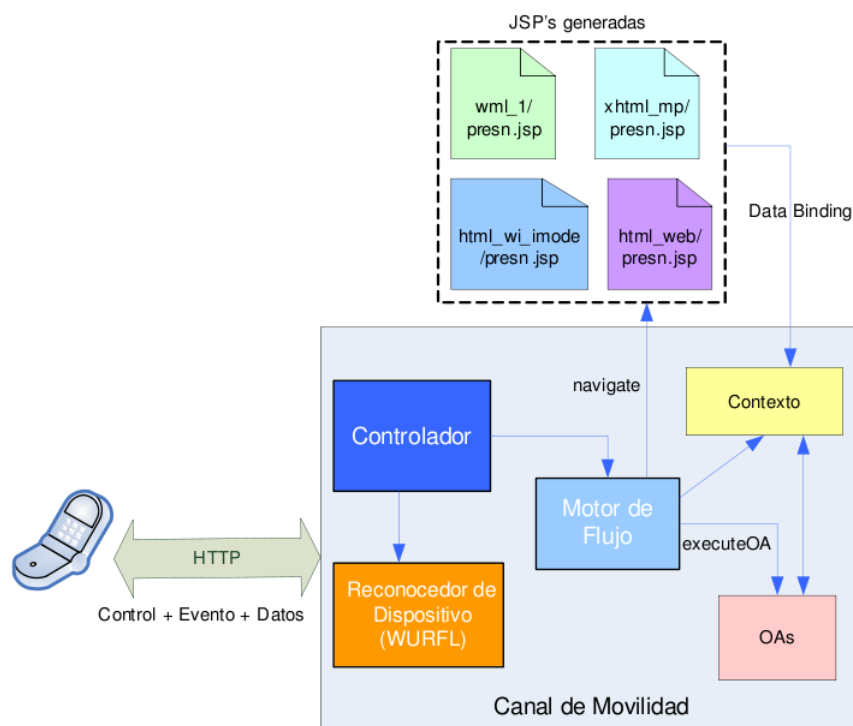


Figura 4.2: Arquitectura d'execució On-Line.

4.5 La capa de Presentació

4.5.1 Definició d'OPs i Presentacions

Aquest és el primer pas a realitzar una es vol construir una nova aplicació. El desenvolupador ha de definir les seves Operacions de Presentació (OPs) que estaran compostes per un conjunt de presentacions amb controls visuals. Aquests controls visuals tindran associat un conjunt d'estils.

Les presentacions poden ser definides independentment del dispositiu o bé per una família de dispositius concreta. Dins de cada OP hi haurà tans subdirectoris com nivells diferents es defineixin per a l'OP (habitualment només el generic, vàlid per a tots els dispositius).

En la figura 4.3 es poden veure les diferents OPs de l'aplicació així com també les presentacions d'una d'elles. Les OPs de Login i Messages es consideren pseudo-ops ja que existiran sempre en totes les aplicacions. L'OP de Login no existirà en cas de que l'aplicació

no requereixi autenticació prèvia. L'OP de Messages s'utilitza per mostrar diferents missatges (warning, error...) en accions showMessage del Flux. El contingut d'aquestes OPs es pot personalitzar per a cada aplicació, i en cas de no fer-ho s'utilitzaran unes per defecte.

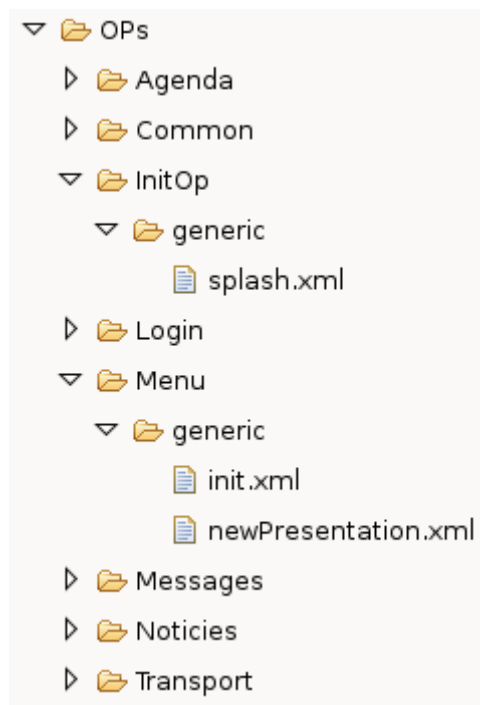


Figura 4.3: Estructura de directoris de les OPs.

El conveni que s'estableix per nombrar OPs i presentacions és que el nom de les OPs comença per majúscules mentre que el nom de les presentacions comença per minúscules.

4.5.2 Estructura de les Presentacions

Les presentacions consten d'una capçalera i un cos. En la capçalera es defineix el títol i els fulls d'estil a aplicar (si n'hi ha). La aparició del títol depen de les capacitats de cada navegador. Si el navegador no el pot mostrar, la plataforma s'encarrega de mostrar-lo en la primera línia de la pantalla.

El cos de les presentacions està format per un conjunt de paràgrafs. Un paràgraf agrupa un conjunt de controls visuals amb una determinada alineació i disposició. La disposició pot ser en vertical, horitzontal o en forma de taula, mentre que l'alineació pot ser a l'esquerra, dreta o centrat.

L'annex 1 mostra una de les presentacions de l'aplicació mòbil per la Fira de Tàrraga.

4.5.3 Definició d'estils

MyMobileWeb adopta l'estàndard CSS per definir els estils dels Controls Visuals en les presentacions. En concret utilitza W-CSS (Wireless CSS). W-CSS afegeix extensions que enriqueixen els fulls d'estil i permeten convertir-los en autèntics fitxers de definició de propietats dels controls visuals.

El principal avantatge de separar les presentacions dels estils està en que és possible redefinir estils per diferents llenguatges de marcat (WML, HTML, XHTML_MP...) o per diferents famílies de dispositius.

Per a redefinir estils només cal crear nous directoris sobre el directori base de CSS. D'aquesta forma els fulls d'estil seran sobreescrits en la corresponent tecnologia i/o família de dispositius. Igual que en les presentacions, els CSS vàlids per totes les tecnologies destí es situaran en el directori generic.

La plataforma MyMobileWeb pot especificar un CSS per defecte, anomenat default.css del qual s'agafen els estils per defecte en cas que el programador no els especifiqui explícitament. Aquests estils estan sintonitzats adequadament per a cada tecnologia.

La figura 4.4 mostra l'estructura de directoris dels fulls d'estil, en aquest cas, amb diferents famílies de dispositius cadascuna amb el seu propi estil.

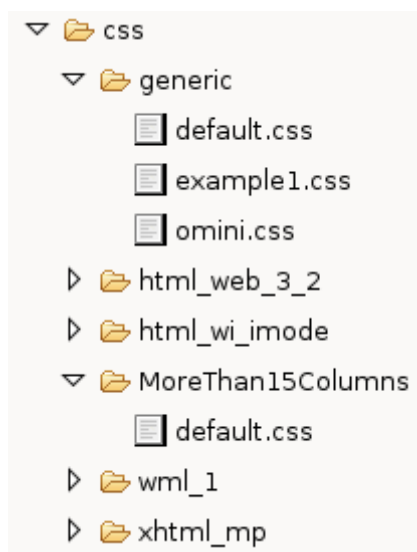


Figura 4.4: Estructura de directoris dels fulls d'estil.

4.5.4 Controls Visuals

MyMobileWeb proporciona una sèrie de controls visuals per a definir les presentacions. L'aspecte i comportament d'aquests s'especifica mitjançant estils W-CSS. Els controls visuals són abstraccions que finalment es converteixen per a una tecnologia i dispositiu concret. Algunes propietats no es poden representar en algunes tecnologies, com seria el cas del color en WAP 1.x.

MyMobileWeb proporciona els següents controls visuals:

- **title:** títol de la pàgina.
- **br:** salt de línia.
- **hr:** línia horitzontal.
- **label:** etiqueta amb text.
- **image:** imatge referenciada a través del catàleg de continguts.
- **link:** enllaç que pot tenir una imatge associada.
- **entryfield:** camp d'entrada de text.
- **datefield:** camp d'entrada d'una data.
- **timefield:** camp d'entrada d'una hora.
- **textarea:** camp de text. És paginable automàticament.
- **select:** llista de valors seleccionable.
- **submit:** control que quan en pulsar-lo provoca una sortida al servidor actualitzant el context.
- **action:** provoca una sortida al servidor però no actualitza el context.
- **reset:** permet esborrar el contingut d'un formulari.
- **menu:** llista d'opcions seleccionables que poden tenir una imatge associada.
- **list:** representa una llista de valors. És paginable automàticament.
- **table:** permet representar un conjunt de valors tabulats. És paginable automàticament.
- **chainedmenu:** conjunt de menús encadenats.
- **upload:** permet pujar fitxers al servidor.
- **telephonecaller:** permet realitzar trucades telefòniques des del navegador del dispositiu, en tots aquells dispositius que ho suportin.
- **phonebookadder:** permet afegir un número de telèfon a l'agenda del dispositiu des del navegador, en tots aquells dispositius que ho suportin.

4.5.5 *Binding* automàtic de dades

MyMobileWeb fa ús intensiu del *binding* automàtic de dades. Aquesta tecnologia permet associar dades amb controls visuals, de forma que el programador no ha de codificar l'associació de les dades amb els controls visuals i aquesta es realitza automàticament per la plataforma.

El *binding* es realitza tant en el sentit control visual - context com en el sentit context - control visual. Per exemple, en un control de tipus taula, la plataforma s'encarrega d'associar una dada del context amb la clau de la fila de la taula. La plataforma emplena automàticament la taula amb les dades del context associades. En cap cas, el desenvolupador ha de realitzar aquestes associacions mitjançant algun tipus de programació. L'única responsabilitat del programador és deixar en el context les dades per poder utilitzar-les en els controls visuals.

Els atributs *bind* i *optionsbind* són utilitzats per realitzar aquest *binding* automàtic. L'atribut *bind* s'utilitza per indicar la variable de context que emmagatzemarà la selecció de l'usuari associada a un control. Per exemple, en un control visual del tipus *select*, l'atribut *bind* contindrà la clau de l'opció seleccionada.

L'atribut *optionsbind* s'utilitza per indicar la variable de context que emmagatzemarà les dades que presentarà el control. En l'exemple anterior, l'atribut *optionsbind* indicaria la variable de context que conté totes les opcions del control.

Binding del Terminal al Context

Com s'ha explicat anteriorment, l'atribut *bind* indica la variable de context que serà actualitzada automàticament amb el valor seleccionat per l'usuari en cada control visual. També apareix un altre atribut, *beantype*, que indica el tipus de JavaBean que contindrà les dades seleccionades per l'usuari i el control visual.

Un cop validades, les dades seleccionades per l'usuari en els controls visuals, van directament al context d'Operació de Presentació. És possible fer que aquestes dades vagin a un altre àmbit del context utilitzant els següents valors:

- **sessionScope**: les dades queden emmagatzemades automàticament en el context de Sessió.
- **presScope**: les dades s'emmagatzemen en el context de Presentació.
- **requestScope**: les dades queden emmagatzemades en el context de Request.
- **applicationScope**: les dades s'emmagatzemen en el context d'Aplicació.
- **opScope**: les dades queden emmagatzemades en el context d'OP (opció per defecte)

Binding del Context al Terminal

La plataforma MyMobileWeb no obliga a treballar amb classes o estructures de dades concretes per fer el *binding* automàtic. D'aquesta manera és possible proporcionar la màxima flexibilitat en el desenvolupament.

Per exemple, el control visual *table* permet a l'atribut *optionsbind* apuntar als següents tipus d'elements:

- Array Java.
- List.
- Map.
- Qualsevol estructura que implementi la interfície `org.morfeo.tidmobile.data.CollectionData`.

Un cop *optionsbind* indica l'estructura que seguirà el control, cal especificar els atributs *member*. En l'exemple de la taula, *keymember* indica el membre de l'estructura de dades que és la clau de cada element de la taula. L'atribut *member* s'associa a cadascuna de les columnes i indica amb quin camp cal omplir-les. Els atributs *member* es poden referenciar per nom o per la posició que ocupen dins de l'array.

4.5.6 JSTL

Les llibreries JSTL són un component dins de l'especificació de Java 2 Enterprise Edition (J2EE) i es controlada per Sun Microsystems. JSTL implementa funcions d'ús freqüent en aplicacions JSP, com ara funcions per manipular cadenes, accés a BD, etc.

MyMobileWeb integra en el seu llenguatge de presentació una sèrie d'etiquetes JSTL 1.1. Amb aquestes etiquetes s'aconsegueix ampliar el llenguatge de referència propi i incrementar la seva potència.

Per a poder utilitzar els tags de JSTL en les presentacions de la plataforma és necessari:

1. Copiar les llibreries `jstl.jar` i `standard.jar` de JSTL 1.1 dins de la carpeta `WEB-INF/lib` de l'aplicació.
2. Especificar el namespace `c` dins del tag `document` en aquelles presentacions que s'utilitzarà JSTL:

```
<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb"
xmlns:c="http://java.sun.com/jsp/jstl/core" id="telefons">
```

A continuació s'expliquen breument les etiquetes que inclou MyMobileWeb:

- **Tag <if>**: etiqueta de condicional simple.
- **Tag <forEach>**: etiqueta d'iteració simple que accepta diferents tipus de col·leccions de dades sobre les que iterar.
- **Tag <choose>**: etiqueta de condició simple que estableix el context d'una exclusió mútua marcada pels tags `when` i `otherwise`.
- **Tag <set>**: etiqueta d'assignació de valors. Assigna el resultat obtingut de l'avaluació d'una determinada expressió a una variable o una propietat d'un objecte.
- **Tag <remove>**: etiqueta d'esborrat. Esborra una variable d'un determinat àmbit si aquest s'especifica.

L'annex 2 mostra una presentació de l'aplicació en la que s'utilitzen tags JSTL.

4.5.7 Desplegament de presentacions

Un cop definides les OPs i les presentacions, cal generar el codi JSP que realitzi les presentacions per a cada tecnologia concreta. El procés de desplegament consisteix en la generació de pàgines JSP, descriptors de validacions i codi de Script:

- Es creen, per cada família, tans JSP's com tecnologies de marcat: `wml_1`, `html_wi_imode`, `html_web_3_2` i `xhtml_mp`.
- Es genera el codi de Script que permet satisfer les validacions especificades en temps de disseny.
- Es generen fitxer `*_meta` que permeten satisfer les validacions especificades en temps de disseny i que recullen la informació necessària per a guardar en el context les dades dels controls visuals.

La figura 4.5 resumeix el procés de definició de la capa de presentació i el seu posterior desplegament.

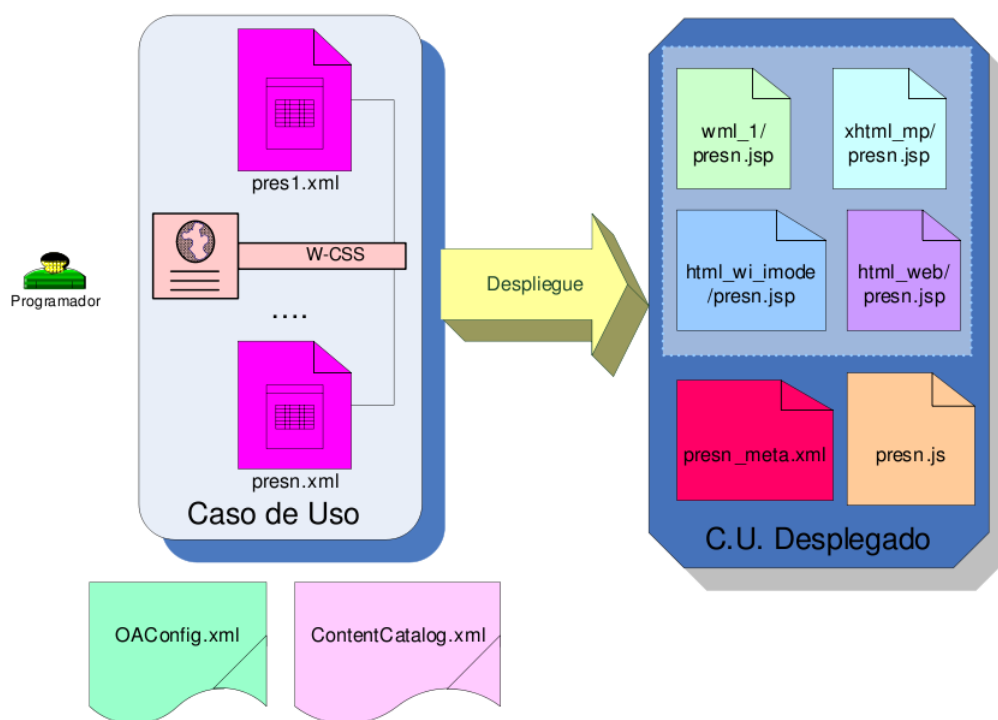


Figura 4.5: Arquitectura de Desenvolupament - Desplegament de presentacions.

4.6 Definició i implementació de OAs

Les Operacions d'Aplicació (OAs) permeten l'obtenció de noves dades i l'execució d'aquelles funcions que sol·licita l'usuari a l'aplicació o sistema. Per exemple: realitzar una consulta a base de dades.

Les OAs obtenen del context les dades necessàries per realitzar la seva funció. Quan obtenen noves dades les deixen sobre el context (sobre les variables que més ens interessin, de cara a realitzar el *binding* automàtic).

Les Operacions d'Aplicació es defineixen mitjançant un fitxer XML en el qual s'indica la classe Java que implementa cada OA. La plataforma MyMobileWeb garanteix que només existeix una instància per cada OA i que a aquesta hi accedeixen n threads. Això cal tenir-ho en compte a efectes de control de concurrència.

4.7 Components

La plataforma MyMobileWeb per a les aplicacions On-Line està formada per una sèrie de components que realitzen l'adaptació de canal en el servidor. La figura 4.6 mostra un diagrama amb la interconnexió dels components i tot seguit es descriu breument el propòsit de cadascun d'ells:

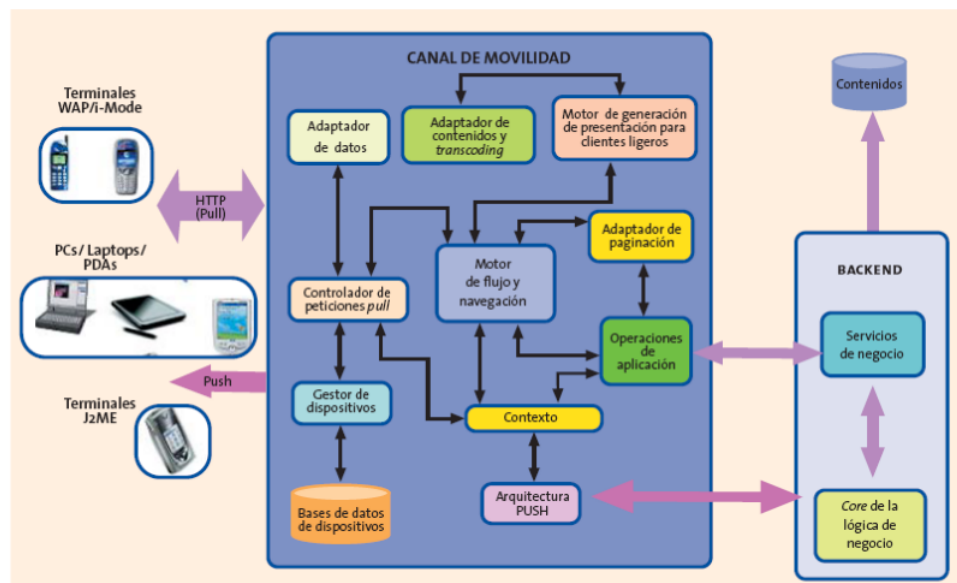


Figura 4.6: Arquitectura d'execució On-Line.

- Gestor de dispositius:
 - Manté una base de dades de dispositius amb la informació de les seves capacitats associades.
 - Realitza el reconeixement de dispositius en temps d'execució.
- Gestor de Condicions i Validacions: s'encarrega de realitzar les validacions automàtiques en el servidor quan el dispositiu no soporta validacions locals (mitjançant llenguatge de script).
- Gestor de Look & Feel i Propietats: s'encarrega d'aplicar les propietats als controls visuals tenint en compte els mecanismes d'herència i overriding.
- Gestor de Context: gestiona dades a diferents nivells jeràrquics utilitzant l'estàndard J2EE de Expression language (EL)
- Gestor de Missatges i Literals: dona suport a les aplicacions multi-idioma.
- Gestor de URL: duu a terme la reescriptura de URLs amb la informació de sessió tenint en compte les capacitats del dispositiu. Els clients sempre naveguen cap a la mateixa URL enviant les dades del context del client, el control i l'event que s'ha produït.
- Motor de Flux: reproduïx el flux de navegació i de crides a Oas.
- Gestor de Paginació per a continguts extens: s'encarrega de dur a terme la paginació automàtica de controls visuals i paràgrafs. Per a cada dispositiu es manté una referència del tamany suportat de pàgina, de manera que les pàgines són fraccionades per no superar aquest lílindar. En el cas de produir-se la paginació automàtica, la plataforma inclou els controls 'Anterior' i 'Següent' per a navegar per les diferents pàgines.
- Catàleg de Continguts: permet definir identificadors abstractes de recursos i una sèrie d'instàncies. En funció de les capacitats del dispositiu s'escolleix la millor instància tenint en compte una sèrie de pesos.

- Adaptador / Traductor de Dades per a SmartClients: gestiona la codificació i decodificació de dades per a la comunicació amb els smartclient J2ME-MIDP.

A continuació s'expliquen més detalladament els components més rellevants de la plataforma MyMobileWeb:

4.7.1 Gestor de Dispositius

El component de Gestió de dispositius és un mòdul fonamental dins d'una plataforma de mobilitat. Les diferents famílies de terminals tenen característiques tècniques molt diverses i això dificulta el desenvolupament d'aplicacions multidispositiu.

La solució a aquests problemes es basa en categoritzar els diferents terminals i emmagatzemar les seves característiques en una base de dades que pugui ser consultada pels programadors i pels components d'adaptació multicanal. En aquesta base de dades també s'hi troba la informació que permet reconèixer els dispositius de l'usuari en temps d'execució.

Conceptes previs:

- Base de Dades de Dispositius: és un repositori de dades que conté informació sobre els diferents terminals mòbils. La informació està estructurada adequadament per permetre un accés eficient. Cada família de dispositius queda determinada per un identificador únic.
- Capacitat: un dispositiu mòbil porta associades una sèrie de característiques, software o hardware, que s'anomenen capacitats. Un exemple de capacitat és: "el terminal disposa d'un navegador que interpreta XHTML-MP".
- Grup de capacitats: les capacitats d'un terminal es poden agrupar en categories. Per exemple, la categoria "display" agruparia a capacitats com la resolució de pantalla, tipus de fonts, mides de les fonts, nombre de files i columnes en pantalla, etc.
- Família de dispositius: una família de dispositius és un conjunt de terminals les capacitats dels quals compleixen una sèrie de requisits. La motivació de definir famílies de dispositius està en que, tot i que existeix una gran varietat de terminals, aquests comparteixen moltes característiques. D'aquesta forma es pot establir un arbre d'herència de famílies.
 - Família de dispositius concreta: una família concreta està constituïda per un conjunt de dispositius que comparteixen el mateix valor en totes les seves capacitats.
 - Família de dispositius abstracta: una família abstracta està constituïda per un conjunt de dispositius que compleixen una sèrie de condicions sobre les seves capacitats. Per exemple, es podria definir la següent família abstracta:

```
resolution_width >=240 and <= 640  
resolution_height >= 320 and <= 480  
available_memory >= 64 Mb and <= 256 Mb
```

L'avantatge principal de disposar de famílies abstractes és que el programador pot definir noves famílies segons convingui, quan les necessitats d'adaptació multidispositiu per una determinada aplicació així ho requereixin.

- **Herència de Famílies:** la idea de les famílies ens duu a l'herència. L'organització dels dispositius es fa en torn a un arbre d'herència. A la part superior de l'arbre es definiran famílies "genèriques" i a la part inferior s'aniran definint famílies més concretes o fins i tot famílies d'un únic dispositiu. L'herència de famílies comporta una herència de les capacitats permetin la redefinició de capacitats.
- **Reconeixement de Dispositius:** és el procés en el qual el Gestor de Dispositius obté tota la informació sobre el terminal amb el que l'usuari es connecta a l'aplicació mòbil. El reconeixement es fa analitzant el contingut de les capçaleres HTTP de la petició realitzada pel dispositiu i utilitzant informació existent a la base de dades de dispositius.

Funcionalitat

El Gestor de Dispositius proporciona la següent funcionalitat:

- **Base de Dades de Dispositius:** el component manté la informació de dispositius, famílies i herència en fitxers XML amb un format conegut. A més, proporciona al programador la possibilitat de sobreescriure valors de capacitats.
- Mitjançant la informació present en la capçalera de la petició HTTP (User-Agent i Accept), es reconeix el dispositiu al que pertany. Un cop s'ha obtingut l'identificador es prova d'emparellar-lo amb una família de dispositius "genèrica" (WAP 1.x, WAP 2.0, i-Mode...). El reconeixement del dispositiu només es sol·licita al crear-se la sessió de l'usuari, emmagatzemant-se aquest identificador per evitar continúes crides per fer la detecció.
- Permet recuperar la informació i característiques del dispositiu a partir de l'identificador d'aquest. La informació del dispositiu es manté en memòria de cara a millorar els temps de recuperació d'aquesta informació.
- Permet consultar la base de dispositius per a conèixer les capacitats i arbre d'herència de diferents models de terminals.

Tecnologia base del component: WURFL

WURFL (Wireless Universal Resource file) va sorgir com un projecte de software obert per a resoldre les carències dels estàndards anteriorment utilitzats (UAPROF i CC/PP). El model de WURFL ha resultat ser àgil i flexible en comparació amb d'altres estàndards que no actualitzen el seu repositori al ritme d'entrada al mercat de nous dispositius. La BBDD de WURFL té informació sobre terminals de diferents mercats: Europa, Àsia, Llatinoamèrica, EEUU...

WURFL proporciona un repositori de dispositius basat en un fitxer XML (wurfl.xml). En aquest fitxer es pot trobar la informació dels dispositius estructurada en forma d'arbre d'herència. Cada node de l'arbre representa un dispositiu físic o tota una família de dispositius. Per a cada dispositiu o família s'associa un identificador, un User-Agent i un identificador de dispositiu pare (fallback). Les parts més altes de l'arbre defineixen famílies "plantilla". Les parts més baixes de l'arbre d'herència defineixen famílies específiques i només es sobreescriuen aquelles capacitats que realment redefeixen. L'arrel de la Base de Dades correspon a un dispositiu anomenat "generic" que estableix totes les capacitats possibles que permet definir WURFL.

WURFL incorpora informació de més de 6000 User-Agent que corresponen a més de 1000 dispositius. La informació de la base de dades de dispositius s'actualitza gràcies a la comunitat d'usuaris de WURFL. Aquesta informació s'estructura en torn a capacitats i grups de capacitats. Existeixen grups de capacitats relacionats amb el navegador, la missatgeria, J2ME, la pantalla, el so, vídeo, descàrregues, etc.

La extensibilitat de WURFL es duu a terme gràcies a l'existència d'un fitxer anomenat wurfl_patch.xml. En aquest fitxer els usuaris de WURFL poden introduir canvis locals (nous dispositius, noves capacitats, rectificacions...) sense tenir que esperar a una nova versió oficial del fitxer wurfl.xml i sense tenir que modificar el fitxer original wurfl.xml. Aquests canvis introduïts en el fitxer poden ser enviats a la comunitat WURFL.

Solució adoptada

WURFL incorpora tota la tecnologia necessària per a resoldre la problemàtica del modelat de la informació de dispositius en base a famílies concretes i herència. Però no incorpora tota la funcionalitat desitjada i dissenyada en el framework MyMobileWeb.

Per això, MyMobileWeb incorpora les següents extensions a WURFL:

- Reconeixement de dispositius analitzant la capçalera accept.
- Noves capacitats que WURFL no inclou (Javascript, disponibilitat de cookies en els navegadors...).
- Possibilitat de definició de més d'un fitxer wurfl_patch.xml.
- Possibilitat de definició de famílies abstractes.
- API Java de Gestió de Dispositius que recobreix al de WURFL i el dota de major funcionalitat.

Procés de reconeixement de dispositius

El procés de reconeixement del dispositiu permet detectar en temps d'execució quin és el dispositiu (o família concreta de dispositius) amb el que l'usuari s'ha connectat a l'aplicació mòbil. El procés de reconeixement es realitza mitjançant la capçalera HTTP de la petició realitzada pel navegador del terminal mòbil. En concret s'utilitzen les capçaleres "User-Agent" i "Accept". La capçalera "User-Agent" conté informació sobre el navegador, fabricant, versió del firmware i altres. D'altra banda, la capçalera Accept conté els tipus MIME que és capaç d'acceptar el terminal mòbil.

En primer lloc s'intenta resoldre el dispositiu a partir de la capçalera "User-Agent". Es tracta de fer coincidir exactament aquest "User-Agent" enviat pel terminal amb un del repositori. En el cas de que no hi hagi cap coincidència exacta, es tracta de buscar coincidències parcials. Finalment, si no hi hagués coincidències parcials es procediria a tractar de reconèixer el dispositiu pel contingut de la capçalera "Accept".

Aquest tipus de reconeixement es basa en associar continguts de la capçalera "Accept" amb famílies concretes de terminals. Aquest reconeixement es fa utilitzant el fitxer `acceptrules.xml`. Per exemple: si la capçalera "Accept" conté el tipus MIME "text/html", llavors es reconeixerà el terminal dins de la família concreta de terminals "HTML". El problema que es pot presentar és que existeixin terminals amb diferents coincidències (per exemple: terminals que suportin continguts WML i HTML). Per això es contempla la especificació de prioritats per aquestes regles, de forma que el terminal serà reconegut tenint en compte les regles de major prioritat.

Arquitectura del component

La figura 4.7 mostra un diagrama d'alt nivell del component de Gestió de Dispositius.

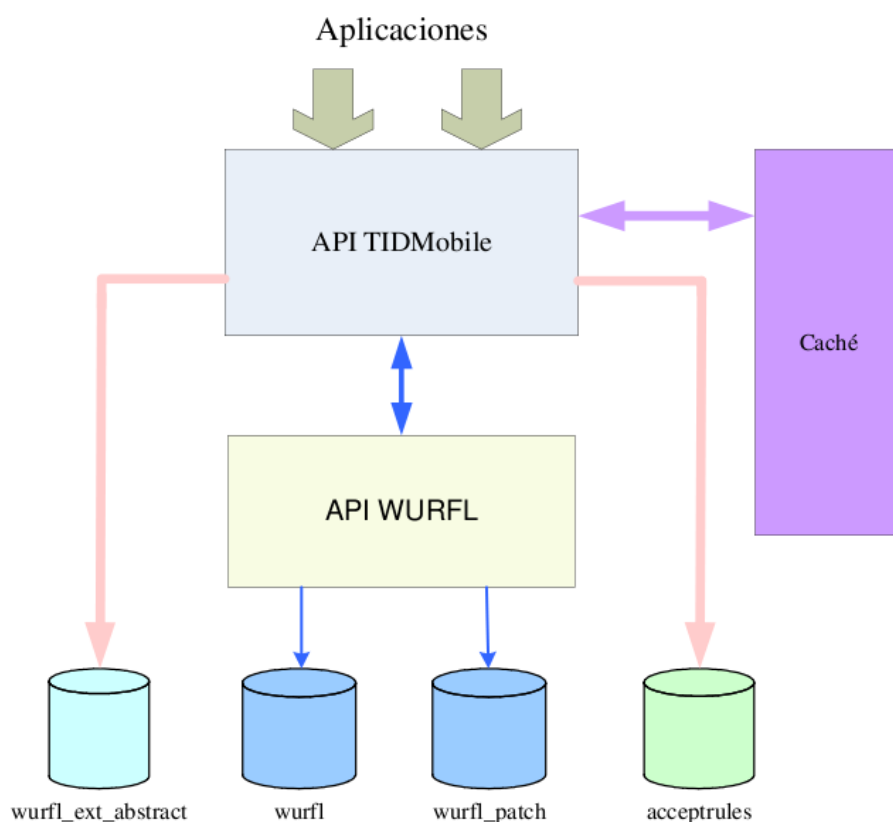


Figura 4.7: Arquitectura del Gestor de Dispositius.

Es proporciona un API d'accés a la funcionalitat per part de les aplicacions. Per la funcionalitat de reconeixement de dispositius i gestió d'informació aquest component crida al API estàndard de WURFL. Pel reconeixement per la capçalera "accept" i la associació amb famílies abstractes, és el propi component el que implementa la seva funcionalitat. Tot això es complementa amb una caché que permet accelerar els processos de detecció de dispositius i de recuperació de la informació sobre aquests.

4.7.2 Gestor de Validacions

La plataforma MyMobileWeb permet especificar les validacions que s'han de realitzar per aquells controls que permeten la introducció d'informació. Un cop especificades aquestes validacions, és la plataforma l'encarregada de dur-les a terme, ja siguin en el servidor o bé el client (si el terminal suporta llenguatge script).

L'especificació de les validacions es realitza combinant atributs i estils dels controls. Mitjançant atributs es defineix els tipus als que seran convertits les dades introduïdes per l'usuari (data, enter, cadena de text...) i mitjançant estils es controlen aspectes com la longitud, la obligatorietat, etc.

Els atributs relacionats amb al conversió automàtica de dades són:

- **validationtype:** especifica el tipus de validació associat. La plataforma validarà que la dada introduïda sigui únicament d'aquest tipus. Els tipus de valors que pot tenir aquest atribut són els tipus nadius de Java (int, short, long, String, Date, Timestamp...)
- **bindingtype:** especifica com s'emmagatzemarà la dada en el context independentment de la validació. Si aquest atribut no s'especifica s'assumeix que el *bindingtype* coincideix amb el *validationtype*. Aquest atribut és útil en el cas que es vulgui validar una dada com un tipus però finalment emmagatzemar-la com un altre tipus. Per exemple, pot interessar que una dada es validi com a enter però s'emmagatzemi com una cadena. En aquest cas s'escriuria: `validationtype="int" bindingtype="String"`.

Les propietats d'estil relacionades amb les validacions són les següents:

- **required: true/false:** indica si l'element és obligatori o no.
- **length:** indica la longitud exacta del camp.
- **maxlength:** indica la longitud màxima del camp.
- **java-date-mask:** indica la màscara d'un tipus data.
- **java-number-mask:** indica la màscara d'un tipus numèric.
- **max-value:** indica el valor màxim. El valor especificat dependrà del tipus de validació.
- **min-value:** indica el valor mínim. El valor especificat dependrà del tipus de validació.

4.7.3 Gestor de Look & Feel i Propietats

MyMobileWeb permet especificar les propietats de les presentacions mitjançant estils W-CSS redefinibles per famílies. Als estils estàndard CSS se'ls afegixen propietats que augmenten les possibilitats funcionals dels controls visuals que utilitza la plataforma.

MyMobileWeb intenta adaptar la definició de presentació especificada pel programador a multitud de dispositius mòbils. Això comporta, degut al gran nombre i varietat de dispositius, que els estils descrits no puguin ser transportats a totes les tecnologies. Així, per exemple, un estil que defineixi un color de fons a una taula només serà aplicat en aquell dispositius que suportin la definició d'un color de fons per les taules.

Els estils més comuns dels controls visuals suportats per MyMobileWeb són:

- **display:** indica si s'ha de mostrar el control.
- **background-color:** color de fons del control.
- **border-style:** estil de les cantonades.
- **margin:** valor de separació respecte a altres controls.
- **padding:** valor de separació entre el contorn del control i el seu contingut.
- **border-color:** color del contorn.
- **border-width:** amplada del contorn.
- **font-family:** tipus de lletra.
- **font-style:** estils de lletra (normal, italic, oblique, h1, h2...)
- **font-variant:** altres variants de font (normal, small-caps, subscript, superscript)
- **font-weight:** grossor de la font (normal, bold, bolder...)
- **text-decoration:** decoració del text (marquee, blink, underlined, line-through)
- **font-size:** mida del text (larger, medium, small)
- **color:** color del text.

Existeixen altres atributs específics per a cada control visual.

4.7.4 Gestor de Context

MyMobileWeb maneja una sèrie de variables de Context especials que es poden referenciar des de les presentacions. Aquestes variables de context representen dades com el perfil d'usuari, dispositiu, idioma, etc. Les variables de context són creades i actualitzades per la plataforma.

Les variables de context són:

- **_MYMW_DEVICE:** dispositiu amb el que es connecta l'usuari. Permet accedir a les capacitats del dispositiu.
- **_MYMW_USER_PROFILE:** emmagatzema el perfil de l'usuari.
- **_MYMW_LOGIN:** emmagatzema el login de l'usuari si l'entrada a l'aplicació es gestiona internament per la plataforma.
- **_MYMW_LANG:** emmagatzema l'idioma de l'usuari. És un String amb la concatenació del codi d'idioma i el codi de país.
- **_MYMW_DEV_CAPS:** és un objecte del tipus Map que permet accedir a les capacitats del dispositiu.

- **_MYMW_DEV_BELONGS:** és un objecte del tipus MAP que permet establir condicions sobre la pertinència del dispositiu de l'usuari a una família abstracta. Per exemple: `${_MYMW_DEV_BELONGS.PdaDevice}` és una expressió que s'evalua com a cert si el dispositiu pertany a aquesta família o fals en cas contrari.
- **_MYMW_DEV_INHERITS:** és un objecte del tipus MAP que permet establir condicions sobre la cadena d'herència del dispositiu de l'usuari.

A més d'aquestes variables, també es pot accedir al Context a través de la classe `org.morfeo.tidmobile.context.Context`. Aquesta classe té els mètodes `getElement` i `setElement` per emmagatzemar i recuperar dades del Context.

També és útil la classe `org.morfeo.tidmobile.server.util.ContextUtil`, que permet recuperar altres dades del Context: OP actual, OP anterior, presentació actual, presentació anterior, dispositiu de l'usuari... Totes aquestes dades es recuperen mitjançant codi Java.

4.7.5 Gestor de Missatges i Literals

Les capacitats multiidioma de la plataforma exigeixen disposar d'aquests dos components que necessiten conèixer en tot moment la situació de les traduccions de literals i de missatges. Ja que la plataforma utilitza aquests components en temps d'execució és necessari configurar els directoris on es troben els missatges i literals.

Per defecte, el directori `i18N` conté tots els fitxers relacionats amb els missatges i literals. Aquests fitxers permeten la internacionalització de les aplicacions.

La figura 4.8 mostra l'estructura de directoris dels literals de l'aplicació. En aquest cas s'han definit tres idiomes: català i castellà i anglès.

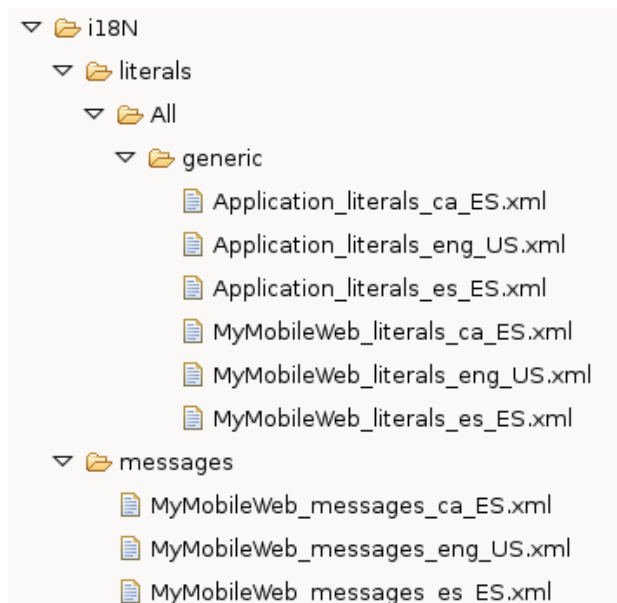


Figura 4.8: Estructura de directoris dels literals.

4.7.6 Motor de Flux

El Gestor de Flux o Gestor de Diàleg és el component fonamental que ha de proporcionar MyMobileWeb. Es tracta d'un component que en tot moment ha de decidir en el servidor quines accions ha d'executar davant d'un event provocat per l'usuari. Les accions a executar dependran del control, de l'event, de l'OP actual, de la presentació actual i en general de les dades presents en el context.

Per facilitar a les aplicacions l'execució d'aquestes accions, MyMobileWeb proporciona un Gestor de Flux basat en events. Les classes manejadores d'events (event handlers) són classes Java normals i corrents, que disposen de mètodes que seran invocats automàticament per MyMobileWeb en quan es produeixi l'event en concret en el terminal mòbil.

Les possibles accions a executar d'una classe manejadora d'events són:

- Navegar cap a una altra presentació (dins de la mateixa OP). La presentació s'especifica amb un nom que serà igual al nom del fitxer XML origen de la presentació.
- Executar una OA.
- Cridar una altra OP.
- Mostrar un missatge.

La figura 4.9 mostra alguns dels events de l'aplicació.

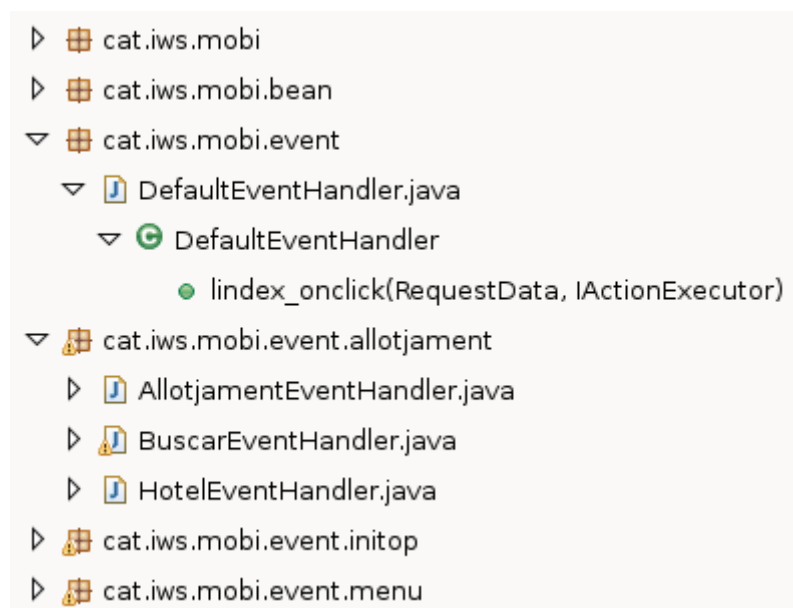


Figura 4.9: Esdeveniments de l'aplicació.

Per exemple, per respondre a l'event onclick del control m1, es codificaria un mètode anomenat *public void m1_onclick(RequestData req, IActionExecutor act) throws Throwable*. Aquest mètode seria invocat automàticament per la plataforma passant-li els paràmetres de forma que el programador només hauria de col·locar el codi de resposta a aquest event (executar una OA, navegar a una altra presentació...). Quan es crida una nova OP s'invoca automàticament el mètode onInitOP.

Quan s'implementa el Gestor de Flux cal tenir en compte la classe RequestData, que proporciona accés al context, al control, a l'event, etc. i la classe ContextUtil, que permet recuperar dades del Context.

4.7.7 Gestor de recursos

Les aplicacions utilitzen recursos, habitualment imatges, que es presenten a l'usuari per fer més amigable la interfície. Quan es desenvolupen aplicacions mòbils cal tenir en compte que no tots els recursos poden mostrar-se en tots els dispositius. Per a mostrar el recurs més adequat per a cada terminal, MyMobileWeb utilitza el concepte abstracte d'identificador de recurs que disposa de diferents instàncies. Per exemple, un recurs "img1" està disponible en jpg, gif i wbmp i en diferents mides.

El catàleg de recursos es defineix mitjançant un sistema de fitxers. Aquest catàleg penja d'una determinada carpeta relativa a la ubicació de recursos. Cada directori dins d'aquesta carpeta representarà un recurs, dins del qual s'hi col·locaran les seves instàncies. L'identificador d'aquest recurs ha de coincidir amb el nom del directori. Per a referenciar un recurs en les presentacions s'utilitza l'atribut resourceid.

Les instàncies aplicables a tots els dispositius es situaran en el directori del recurs o bé en la subcarpeta generic. Aquelles instàncies específiques d'una determinada família de dispositius o d'una determinada tecnologia s'allotjaran en una subcarpeta el nom de la qual serà el de la família o tecnologia indicada. Per exemple, una instància aplicable tan sols a dispositius wml_1 es situarà dins de la subcarpeta wml_1.

En el moment de decidir la instància a enviar a cada dispositiu, la plataforma agafarà aquella que per les seves característiques es visualitzi millor en el dispositiu. Aquesta instància s'agafarà d'aquella subcarpeta que correspongui a la família o tecnologia del dispositiu. Si aquesta no existeix, es buscarà la millor instància entre totes les definides pel recurs.

El desenvolupador té la responsabilitat de definir en el sistema de fitxers de recursos aquestes instàncies. Posteriorment, la plataforma s'encarregarà d'enviar la millor instància a cada terminal.

La figura 4.10 mostra l'estructura de directoris dels recursos.

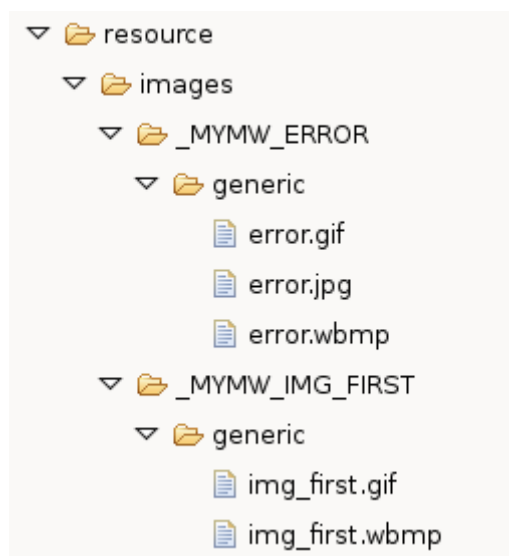


Figura 4.10: Estructura de directoris dels recursos.

4.7.8 Transcodificador d'imatges

MyMobileWeb disposa d'un component de transcodificació d'imatges anomenat MyMobileWeb Transcoder. Les seves funcions són les següents:

- Redimensionar les imatges en funció de les mides de la pantalla.
- Convertir el format per adaptar-se al tipus d'imatge que suporta cadascun dels dispositius destí.

Per aplicar el transcodificador d'imatges s'utilitzen les següents propietats d'estil (en els fitxers .css):

- **Transcode**: valor booleà que indica si cal aplicar o no la transcodificació sobre una determinada imatge.
- **weight-width**: amplada que ha de tenir la imatge respecte l'amplada de la pantalla.
- **weight-height**: altura que ha de tenir la imatge respecte l'altura de la pantalla.

4.8 Configuració de l'aplicació en el servidor

La plataforma MyMobileWeb s'ha de configurar mitjançant els següents fitxers XML:

- **MyMobileWeb.DevMng.xml**: conté la configuració del component DeviceManager, que incorpora la base de dades amb les capacitats dels dispositius.
- **MyMobileWeb.PropMng.xml**: conté les rutes dels diferents fulls d'estils a aplicar.
- **MyMobileWeb.ImageTranscoder.xml**: conté la configuració del transcodificador d'imatges.
- **OACfg.xml**: conté la declaració de les OAs que s'utilitzen en l'aplicació.
- **MyMobileWeb.Global.xml**: conté la configuració dels diferents components d'ús comú en l'arquitectura.

Adicionalment, és possible especificar un fitxer XML de configuració específic per l'aplicació que ha de seguir el mateix format que els altres fitxers: MyMobileWeb.xxx.xml. La situació dels fitxers de configuració s'especifica mitjançant un paràmetre de context en el fitxer web.xml associat a l'aplicació que utilitza la plataforma.

Aquests són alguns dels paràmetres configurables més importants del fitxer MyMobileWeb.Global.xml:

- Definició de la primera OP que s'executarà en l'aplicació:

```
<property name="First_Application_OP" value="InitOp">
```

- Definició del Gestor de Login:

- NoLogin: per aquelles aplicacions que no requereixen una pàgina d'autenticació.
- RemoteUserLogin: quan l'autenticació es resol mitjançant un mecanisme tipus "Remote User" (autenticació bàsica)
- CustomLogin: utilitzat habitualment en aplicacions amb pàgina de login pròpia.
- CustomLoginHeader: el login s'agafa d'una capçalera. S'utilitza principalment quan l'identificador d'usuari és el número de telèfon des del que es realitza la connexió.
- CustomLoginLDAP: és semblant a CustomLogin però realitza la validació de l'usuari contra un servidor LDAP.

```
<module name="LoginManager">  
  <property name="Login_Manager" value="org.morfeo.tidmobile.server.login.NoLogin"/>  
</module>
```

- Definició del llenguatge per defecte de l'aplicació:

```
<property name="Default_Language" value="es_ES"/>
```

- Especificació dels llenguatges de l'aplicació:

```
<property name="Installed_Languages" value="eng_US,es_ES"/>
```

En l'annex 3 es mostra el fitxer de configuració MyMobileWeb.Global.xml de l'aplicació mòbil creada per la Fira de Tàrraga.

4.9 Procés de desenvolupament

El desenvolupament de les aplicacions de mobilitat (noves o basades en la mobilització d'una ja existent) és semblant al desenvolupament de qualsevol aplicació Web J2EE, tenint en compte que cal afegir al directori WEB-INF/lib les llibreries corresponents a la plataforma de mobilitat. Utilitzant el plugin de MyMobileWeb per a Eclipse, aquestes llibreries s'afegeixen automàticament.

El desenvolupament d'una aplicació mòbil utilitzant la plataforma MyMobileWeb s'ha de realitzar seguint els següents passos:

1. Anàlisis dels requeriments de l'aplicació i divisió de la funcionalitat en OPs (casos d'ús).
2. Per a cada una de les OPs, especificar la capa de presentació mitjançant fitxers XML de definició de vistes i fulls d'estil CSS.
3. Definició de les validacions d'entrada de dades mitjançant propietats dels controls i atributs.
4. Definició del *binding* dels controls amb les dades del context.
5. Creació del catàleg de continguts amb les imatges a visualitzar en l'aplicació.
6. Definició del flux d'accions a realitzar davant dels events provocats per l'usuari.
7. Definició i implementació de les OAs que obtindran noves dades i realitzaran operacions d'usuari.
8. Configuració dels components de la plataforma en el servidor Web.
9. Desplegament de les presentacions i de l'aplicació en el contenidor Web.

Capítol 5

Cas pràctic: Fira de Tàrrega

5.1 Descripció del sistema existent

En aquesta secció no s'intenta explicar cap qüestió tècnica sobre el portal de la Fira de Tàrrega, la seva plataforma d'execució o les tecnologies utilitzades pel seu desenvolupament, sinó que s'explica com estan organitzats els seus continguts.

El portal Web de la Fira de Tàrrega es pot consultar en tres idiomes: Català, Castellà i Anglès. Els continguts més importants estan organitzats en diferents seccions:

- Informació útil:
 - Ajuntament de Tàrrega.
 - Plànol de la ciutat.
 - Allotjament.
 - Zona d'acampada.
 - Adreces d'interès.
 - Enllaços.
 - Preguntes freqüents.
 - Contacte.
- Informació de la fira:
 - Què és la fira.
 - Venta d'entrades.
 - Patronat.
 - Crèdits.
 - Història.
 - Xarxes.
 - Notícies.
- Àrea professionals:
 - Com participar.
 - La Llotja.
 - La Llotja virtual.

- Àrea artística:
 - Com participar.
 - Condicions.
 - Promoció a Tàrraga.
- Àrea premsa:
 - Dossiers informatius.
 - Material gràfic.
 - Acreditacions.
- Àrea recursos:
 - Patrocinadors i col·laboradors.
 - Publicitat.
 - Merchandising.
 - Ocupació de la via pública.
 - Mercat de paradistes.

5.2 Portal Web mòbil de la Fira de Tàrraga

Després de reunir-nos amb responsables de la Fira de Tàrraga, s'ha cregut convenient oferir en el portal Web mòbil els següents serveis:

- Consulta de la programació de la fira.
- Consulta de telèfons i adreces d'interès.
- Búsqueda d'allotjament.

El context en que es trobarà l'usuari que accedeixi al portal Web de la Fira de Tàrraga és clau. La informació desitjada serà molt més concreta i específica. Els continguts del portal Web mòbil han de ser senzills però funcionals.

De la mateixa manera que en la Web actual, el portal mòbil ha d'estar disponible en els mateixos tres idiomes: Català, Castellà i Anglès. MyMobileWeb té un component que permet extreure els literals de l'aplicació i facilitarà aquesta tasca.

Pel que fa a la informació de la base de dades, alguns camps estan traduïts i son els que es mostren en l'idioma corresponent. Cal dir que la tasca d'introduir la informació a la base de dades i traduir-la correspon a l'organització de la Fira de Tàrraga. És per això que no s'ha pogut dur un control d'errors en aquest aspecte.

5.3 Desenvolupament de l'aplicació

5.3.1 Casos d'ús

El terme cas d'ús s'utilitza en enginyeria del software. Un cas d'ús és una tècnica per a capturar els requisits potencials d'un nou sistema. Els diagrames de casos d'ús serveixen per identificar la funcionalitat, el comportament del sistema i com ha de ser la interacció amb l'usuari. Els casos d'ús estan expressats des del punt de vista de l'actor, que és el rol que juga l'usuari respecte al sistema.

La figura 5.1 mostra els casos d'ús que s'han identificat per aquesta aplicació.

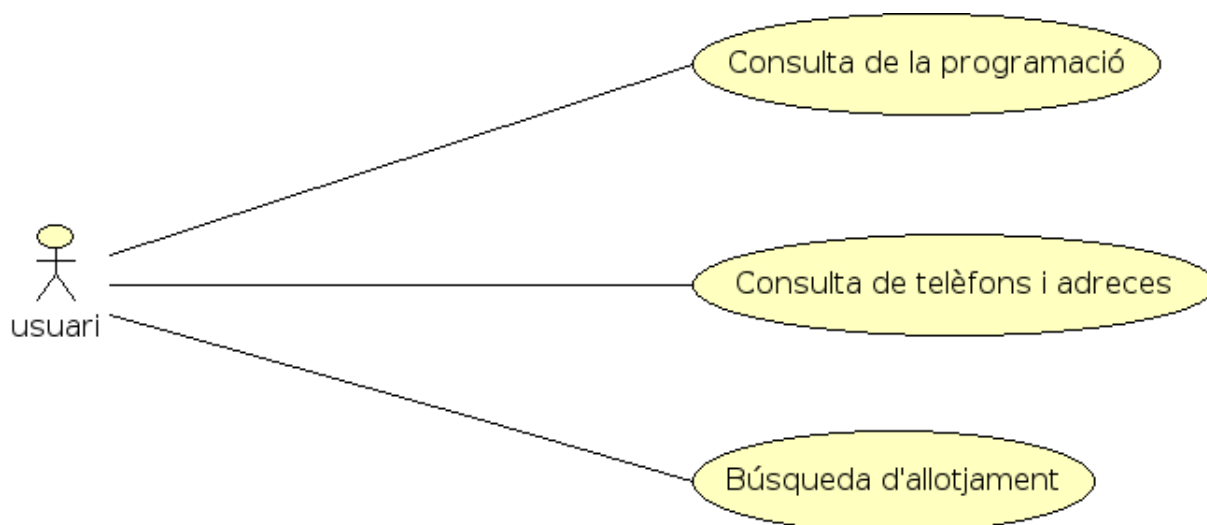


Figura 5.1: Diagrama de casos d'ús.

A continuació es descriuen els diferents casos d'ús que contempla l'aplicació mòbil:

Consulta de la programació

- **Propòsit:** l'usuari desitja consulta la programació d'espectacles de la Fira de Tàrraga.
- **Precondició:** l'usuari ha accedit a l'aplicació mitjançant un terminal mòbil.
- **Postcondició:** el sistema no canvia d'estat.
- **Seqüència normal d'esdeveniments:**
 1. L'usuari indica que vol consultar la programació d'espectacles seleccionant aquesta opció en el menú principal de l'aplicació.
 2. El sistema presenta la possibilitat de cercar els espectacles alfabèticament, segons el dia, per companyia o bé per l'espai on es realitzen.
 3. L'usuari selecciona una de les opcions anteriors.
 4. El sistema mostra mitjançant un conjunt de menús encadenats els espectacles filtrant-los a partir de l'opció seleccionada.
 5. L'usuari selecciona un espectacle.
 6. El sistema mostra al visitant la informació principal d'aquest espectacle i permet accedir a una altra presentació amb més informació.

Búsqueda d'allotjament

- **Propòsit:** l'usuari desitja cerca allotjament.
- **Precondició:** l'usuari ha accedit a l'aplicació mitjançant un terminal mòbil.
- **Postcondició:** el sistema no canvia d'estat.
- **Seqüència normal d'esdeveniments:**
 1. L'usuari indica que vol cercar allotjament seleccionant aquesta opció en el menú principal de l'aplicació.
 2. El sistema presenta la possibilitat de filtrar la informació per les diferents localitats.
 3. L'usuari selecciona una localitat.
 4. Si hi ha més d'un allotjament en aquella localitat es mostra un llistat. En cas d'haver-hi un únic allotjament en la localitat seleccionada, automàticament es mostra la informació d'aquest (pas nº 6).
 5. L'usuari selecciona una opció del llistat anterior.
 6. El sistema mostra al visitant les dades més important de l'allotjament seleccionat. Si l'usuari està accedint amb un dispositiu de tipus PDA el sistema mostra més informació (descripció i fotografia).

Consulta de telèfons i adreces

- **Propòsit:** l'usuari desitja consultar les adreces i telèfons d'interès.
- **Precondició:** l'usuari ha accedit al sistema mitjançant un terminal mòbil.
- **Postcondició:** el sistema no canvia d'estat.
- **Seqüència normal d'esdeveniments:**
 1. L'usuari indica que vol consultar les adreces i telèfons d'interès seleccionant aquesta opció en el menú principal de l'aplicació.
 2. El sistema mostra les adreces i telèfons d'interès de forma paginada.

5.3.2 Divisió de la funcionalitat en OPs

A partir dels casos d'ús anteriors, s'han creat les següents Operacions de Presentació per l'aplicació:

- **Allotjament:** aquesta OP permet a l'usuari consultar l'allotjament.
- **Telèfons:** aquesta OP permet a l'usuari consultar els telèfons d'interès.
- **Programació:** aquesta OP permet a l'usuari consultar la programació de la fir
- **InitOp:** és l'OP inicial de l'aplicació. Correspon a una pantalla de benvinguda on es pot seleccionar l'idioma.
- **Menu:** aquesta OP correspon al menu principal de l'aplicació que permet navegar a les diferents seccions.
- **Common:** aquesta OP conté elements comuns a totes les presentacions com la capçalera i el peu de pàgina.
- **Messages:** és una OP gestionada internament per MyMobileWeb que mostra els missatges de l'aplicació (errors, advertències...)

5.3.3 Descripció de les presentacions

Totes les OPs defineixen les vistes dins del nivell *generic* i són multidispositiu. A continuació es descriuen les presentacions utilitzades per l'aplicació, la seva funcionalitat i característiques principals:

- **Allotjament/generic/buscar.xml:** especifica la presentació que permet buscar allotjament. En aquesta presentació s'han utilitzat menus encadenats. La búsqueda es realitza per ciutats i després per allotjaments dins d'aquella ciutat (si n'hi ha més d'un).
- **Allotjament/generic/hotel.xml:** defineix la presentació que mostra un allotjament en concret. S'empra el control visual *telephonecaller* que permet trucar automàticament al número de telèfon. En el cas d'un dispositiu tipus PDA o PC es mostra més informació: descripció, mail, url i una fotografia.
- **Common/generic/template.xml:** representa una presentació amb tres paràgrafs: un per l'encapçalament i dos per diferents peus de pàgina. Aquests paràgrafs s'inclouen a la resta de presentacions segons convingui.
- **InitOp/generic/splash.xml:** defineix la presentació inicial que presenta l'aplicació. Inclou el logotip de la Fira de Tàrraga i permet canviar d'idioma.
- **Menu/generic/menu.xml:** especifica la presentació amb el menú principal de l'aplicació. Cada element del menú té una imatge associada i un atribut *longtitle* amb un text explicatiu. Aquest text només es mostra en els terminals amb major capacitat visual.
- **Messages/generic/error,warning,information,question,plain.xml:** defineix les presentacions amb missatges per a mostrar a l'usuari. Són presentacions controlades per la plataforma però que en aquest cas s'han editat canviant el seu aspecte.
- **Telefons/generic/telefons.xml:** mostra els telèfons d'interès. En aquesta presentació s'han utilitzat tags JSTL.
- **Programacio/generic/buscar.xml:** correspon a una presentació amb un menú que permet triar com es vol realitzar la búsqueda.
- **Programacio/generic/buscar_companyia.xml:** permet realitzar una búsqueda de la programació segons la companyia.
- **Programacio/generic/buscar_dia.xml:** permet veure els espectacles que es realitzen en un dia en concret.
- **Programacio/generic/buscar_espai.xml:** permet realitzar una búsqueda de la programació segons l'espai on es fa l'espectacle.
- **Programacio/generic/buscar_espectacle.xml:** permet consultar els espectacles alfabèticament.
- **Programacio/generic/espectacle.xml:** defineix la presentació que mostra informació de l'espectacle seleccionat.
- **Programacio/generic/espectacle_mesinfo.xml:** especifica la presentació que mostra més detalls sobre un espectacle en concret. S'hi accedeix des de la presentació anterior.

5.3.4 Descripció dels estils

L'aplicació fa ús de la redefinició d'estils per personalitzar l'aparença i comportament dels controls visuals en funció de les diferents famílies de dispositius. L'estructura dels fulls d'estil CSS és la següent:

- **generic/firatarrega.css:** conté els estils de l'aplicació comuns per a totes les tecnologies i dispositius.
- **PdaDevice/firatarrega.css:** conté els estils redefinits per a la família de dispositius PDA.
- **PcDevice/firatarrega.css:** conté els estils redefinits per si l'usuari accedeix al portal mòbil amb un ordinador convencional.

5.3.5 Descripció de les OAs

L'aplicació disposa de diferents OAs que calculen les dades que s'han de mostrar en les diferents presentacions. Aquestes OAs estan definides dins del paquet `cat.iws.mobi.oa`. A continuació es descriuen:

- **GetTelefons:** aquesta OA accedeix a la taula telefons de la BD i deixa les dades en el context. Aquestes dades són utilitzades en la presentació `telefons.xml`.
- **Chmenu1_m1_OA:** aquesta OA està relacionada amb el menú encadenat que permet la búsqueda d'allotjament. En concret obté les ciutats. El nom de l'OA ve determinat per l'identificador del control.
- **Chmenu1_m2_OA:** aquesta OA també està relacionada amb el menú encadenat que permet la búsqueda d'allotjament. En concret els diferents allotjaments d'una ciutat. El nom de l'OA ve també determinat per l'identificador del control.
- **GetHotel:** aquesta OA obté les dades de l'allotjament seleccionat i les deixa en el context. Aquestes dades són utilitzades en la presentació `hotel.xml`.
- **Chmenu2_m1_OA, Chmenu3_m1_OA, Chmenu3_m2_OA, Chmenu4_m1_OA, Chmenu4_m2_OA, Chmenu5_m1_OA, Chmenu5_m2_OA:** aquestes OAs corresponen a l'OP de Programació i permeten obtenir els diferents espectacles segons el dia, l'espai d'actuació, la companyia o alfabèticament.
- **GetEspectacle:** aquesta OA es crida un cop s'ha seleccionat un espectacle en concret. Obté les dades de l'espectacle i les deixa en el context.

5.3.6 Descripció de les classes del control de flux

Les classes amb el control de flux pengen totes de la ruta `cat.iws.mobi.event`. Hi ha diferents subpaquets, un per cada OP que s'encarreguen del maneig dels events de cada OP i presentació:

- **cat.iws.mobi.event:** s'encarrega del maneig dels events comuns en totes les OPs i presentacions. En aquest cas, de l'event que es produeix en fer click sobre `index`.
- **cat.iws.mobi.event.allotjament:** aquesta classe s'encarrega del maneig dels events de l'OP Allotjament i de les seves presentacions. Les OAs associades al control `Chmenu1` són cridades automàticament per la plataforma. Cal destacar l'event `Chmenu1_onfinish` que crida a l'OA `GetHotel` i navega cap a la presentació `hotel.xml`.

- **cat.iws.mobi.event.initop:** és la classe que s'encarrega del maneig dels events de l'OP InitOP i de la seva presentació splash.xml.
- **cat.iws.mobi.event.menu:** s'encarrega del maneig dels events de l'OP Menu i de la presentació menu.xml. L'event menu1_onclick s'encarrega de navegar cap a les diferents presentacions segons l'opció del menú seleccionada.
- **cat.iws.mobi.event.programacio:** s'encarrega del maneig dels events de l'OP Programació i de les seves presentacions.
- **cat.iws.mobi.event.telefons:** s'encarrega del maneig dels events de l'OP Telèfons i de la presentació telefons.xml.

5.3.7 Estructura de directoris de l'aplicació

- **config:** conté els fitxers de configuració de l'aplicació (MyMobileWeb.Global.xml, MyMobileWeb.DevMng.xml, MyMobileWeb.ImageTranscoder.xml, OACConfig.xml). També es troba un subdirectori logs que conté el fitxer de configuració de logs per l'aplicació.
- **css:** conté els fulls d'estil que defineixen l'aspecte de les presentacions i controls visuals.
- **DeviceBBDD:** conté tots els fitxers relacionats amb el Gestor de Dispositius i WURFL. El fitxer wurfl_ext_abstract_1.xml conté la definició de les famílies abstractes que s'han utilitzat en l'aplicació.
- **i18N:** conté els fitxers relacionats amb missatges i literals. Aquests fitxers permeten la internacionalització de l'aplicació.
- **OPs:** conté l'especificació de les Operacions de Presentació i les seves vistes associades (fitxers .xml).
- **jsp:** conté les pàgines JSP generades automàticament a partir de les OPs, les presentacions definides en XML i els fulls d'estil CSS.
- **resource:** aquest directori conté les imatges utilitzades en l'aplicació.
- **temporal:** directori temporal que conté les imatges transcodificades amb el component ImageTranscoder de la plataforma.
- **script:** conté els fitxers de script necessaris per a realitzar les validacions. Aquests fitxers es generen automàticament juntament amb les presentacions.
- **WEB-INF:** directori estàndard que conté el fitxer descriptor de l'aplicació web.xml, el directori amb les classes i les llibreries necessàries per a l'execució. També conté els fitxers de configuració i de mapeig de Hibernate.

Capítol 6

Proves de validació de l'aplicació

6.1 Validador ready.mobi

El validador de ready.mobi avalua un lloc Web per a mòbils segons els estàndards definits. Aquest informe dona una puntuació entre 1 i 5 i analitza la viabilitat de la Web en dispositius mòbils.

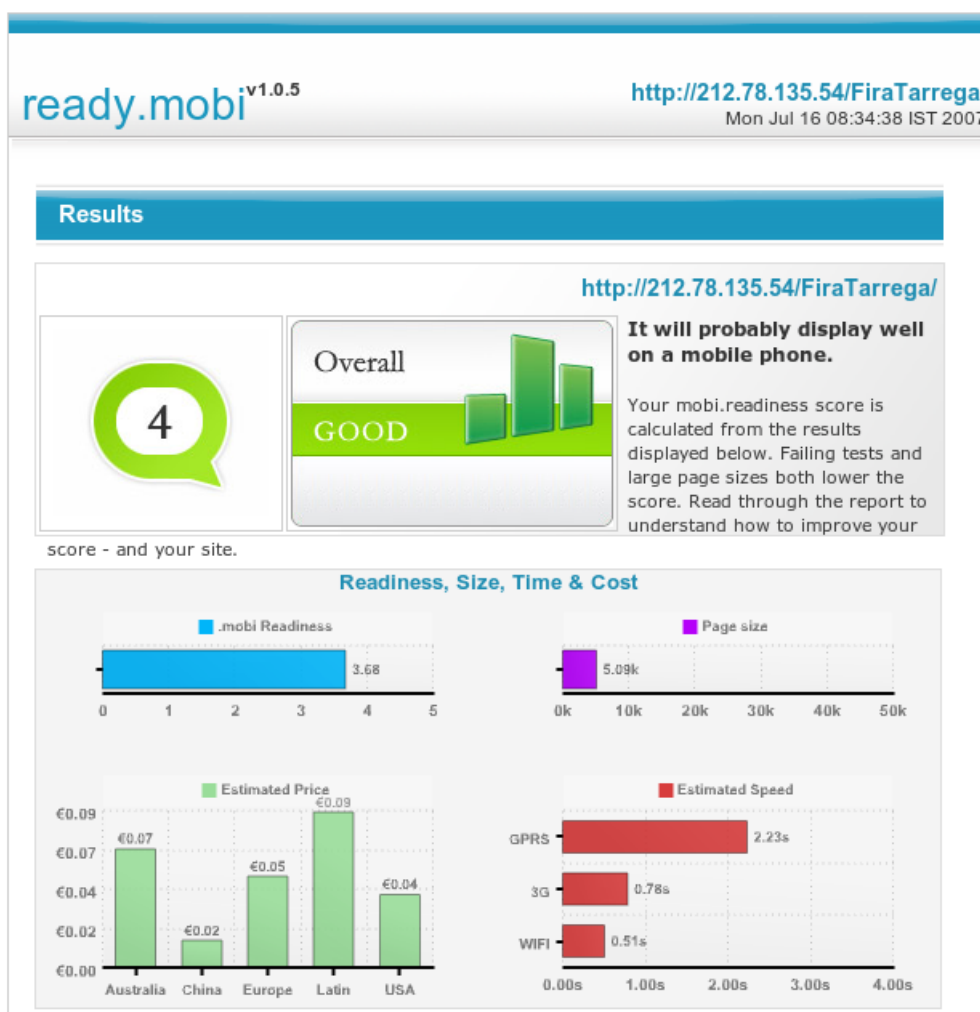


Figura 6.1: Validador ready.mobi.

6.2 Proves amb emuladors

L'aplicació mòbil s'ha provat amb èxit utilitzant els següents emuladors:

6.2.1 Nokia N70



Figura 6.2: Proves amb Emulador Nokia N70 - 1.



Figura 6.3: Proves amb Emulador Nokia N70 - 2.

6.2.2 Sony Ericsson k750i



Figura 6.4: Proves amb Emulador Sony Ericsson k750i - 1.



Figura 6.5: Proves amb Emulador Sony Ericsson k750i - 2.

6.2.3 Openwave V7 Simulator



Figura 6.6: Proves amb Emulador Openwave V7 Simulator - 1.

6.2.4 Motorola v3i



Figura 6.7: Proves amb Emulador Motorola v3i - 1.

6.2.5 Sharp GX-10



Figura 6.8: Proves amb Emulador Sharp GX-10 - 1.

Capítol 7

Conclusions i possibles millores

7.1 Conclusions

Amb l'arribada del domini .mobi i el creixent ús dels telèfons mòbils per a tasques fins ara impensables, les pàgines Web orientades a dispositius mòbils han experimentat un gran creixement que es preveu continuat. Igualment, La diversitat de dispositius i tecnologies mòbils continuarà creixent.

En un futur pròxim, l'usuari utilitzarà un dispositiu mòbil, principalment el telèfon, com el seu mitja primari d'accés a Internet. Els telèfons mòbils són més barats que un PC, n'hi ha més del triple, i el seu creixement és el doble de ràpid. La Web mòbil serà el que predomini en la major part de països arreu del món, amb dispositius i xarxes cada cop més potents.

Aquesta evolució obligarà a un nou replantejament del disseny Web: les funcionalitats i continguts bàsics hauran de ser accessibles des de qualsevol tipus de dispositiu, navegador o connexió; mentre que pels dispositius més avançats es podran mostrar altres tipus de continguts.

Personalment, realitzant aquest treball he pogut ampliar els meus coneixements en molts àmbits més enllà del desenvolupament d'aplicacions per a mòbils que no han quedat plasmats en aquest document. Tot això queda per mi i és el que en el fons importa.

Conviure durant aquests mesos amb tot tipus de dispositius mòbils m'ha fet adonar de la importància de la telefonia mòbil en el futur d'Internet i de la Web. He quedat molt sorprès en veure algunes de les estadístiques i dades referent als telèfons mòbils del segon capítol. Navegar des d'un telèfon mòbil és encara un servei poc utilitzat, però en pocs anys, la majoria de dispositius tindran accés a Internet de forma més ràpida, fàcil i econòmica gràcies a tecnologies com Wi-Fi.

Per llavors, hem de ser capaços de mobilitzar la Web i fer-la accessible des de tots aquests dispositius. Avui en dia encara falta molta estandarització respecte a les tecnologies que permeten l'accés a Internet des de dispositius mòbils i hi ha moltes diferències entre els diferents fabricants. Utilitzar una plataforma com MyMobileWeb permet oblidar-se d'aquesta heterogeneïtat i fer un disseny multidispositiu.

Aquest és només un primer pas de cara al futur d'Internet, amb una Web unificada per a tot tipus de dispositius. Cal dissenyar un mateix lloc Web per a diverses plataformes, però no amb els mateixos continguts, sinó que cal adaptar-los a cadascuna d'elles.

A continuació es destaquen les conclusions més destacades a les que s'ha arribat després de realitzar aquest treball:

Beneficis de la Web mòbil

La Web mòbil permetrà accedir a la informació en qualsevol moment i en qualsevol lloc on hi ha cobertura telefònica. Alliberar la Web d'aquestes restriccions de mobilitat permetrà un intercanvi major d'informació.

La Web mòbil obrirà noves oportunitats de negoci, oci i altres serveis que permetran augmentar la nostra capacitat de moviment.

Futur de la telefonia mòbil

La quarta generació de telefonia mòbil suposarà un nou pas de les telecomunicacions cap a la mobilitat i accessibilitat universals. La integració de noves tecnologies com Wi-Fi, sistemes de localització, reconeixement de veu, pantalles tàctils... augmentarà el ventall de possibilitats de dels dispositius mòbils.

En molts països, com el Regne Unit, Itàlia o Espanya, hi ha més telèfons mòbils que habitants. A finals de 2007 es preveu que el nombre d'usuaris de telèfons mòbils superi els 3000 milions, equivalent a la meitat de la població mundial.

Polivalència de la plataforma MyMobileWeb

Emprar la plataforma MyMobileWeb ens ha facilitat l'adaptació del portal Web a dispositius mòbils. Els seus controls visuals dissenyats per a mobilitat permeten mostrar qualsevol tipus d'informació en qualsevol tipus de dispositiu mòbil.

A més a més, la plataforma permet realitzar aplicacions off-line o aplicacions que no necessiten una connexió on-line permanent.

Importància del Programari Lliure

Realitzar aquest treball en una empresa que basa el seu desenvolupament en el programari lliure m'ha permès veure la seva importància. Totes les eines utilitzades per a la realització d'aquest treball són de lliure distribució: des de l'eina principal de desenvolupament utilitzada, Eclipse; fins a la plataforma MyMobileWeb, basada en tecnologies i estàndards oberts.

Sense el Programari Lliure la realització d'aquest treball hauria estat molt més complicada i econòmicament inviable.

7.2 Treball futur i possible millores

En aquest treball només s'empren algunes de les possibilitats de MyMobileWeb. En futures versions de la plataforma s'inclouran nous controls visuals que suposaran una millora visual i d'usabilitat de les aplicacions. Entre d'altres s'espera un nou control visual que permetrà la integració de MyMobileWeb amb sistemes de localització GPS i una millora en la transcodificació i redimensió d'imatges.

Tot i les proves de validació amb diferents dispositius i emuladors, és possible que amb alguns d'ells la visualització i/o navegació no sigui l'esperada. Per solucionar aquests problemes es podrien redefinir els fulls d'estils CSS per a famílies de dispositius en concret. També seria convenient mantenir actualitzada la versió de la base de dades de WURFL ja que constantment apareixen nous models de telèfons mòbils.

Una de les millores més importants seria assignar a cada enllaç un AccessKey. Això milloraria l'accessibilitat i la navegació de l'aplicació i permetria seguir els enllaços amb el teclat numèric del dispositiu. En la nostra aplicació només s'han assignat AccessKeys al menú principal i al menú de l'OP de Programació.

Finalment, tot i disposar d'una nova direcció: mobi.firatarrega.com, és possible que molts usuaris segueixin accedint a www.firatarrega.com. Al ser un servidor completament diferent, MyMobileWeb no se n'encarrega. Seria convenient redireccionar aquestes peticions fetes amb dispositius mòbils.

Capítol 8

Annexes

8.1 Annex 1 - Exemple d'una presentació en .xml

OPs/InitOp/generic/splash.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="firatarrega.css" type="text/css"?>
<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb" id="splash">
  <mymw:head>
    <mymw:title></mymw:title>
  </mymw:head>
  <mymw:body>

    <mymw:p id="p1" align="center">
      <mymw:image id="img1" alt="Fira de Tàrrega" src="resource/images/logo/logo.jpg"
        style="logo"/>
    </mymw:p>

    <mymw:p id="p2" align="center">
      <mymw:link style="blackbold" id="entrar">Entrar</mymw:link>
    </mymw:p>

    <mymw:br/>
    <mymw:p id="p3" style="mlang">
      <mymw:link id="ca_ES" style="img" resourceid="catala"
        href="DH?_MYMW_LANG=ca_ES">Català</mymw:link>
      <mymw:link id="es_ES" style="img" resourceid="spain"
        href="DH?_MYMW_LANG=es_ES">Español</mymw:link>
      <mymw:link id="eng_US" style="img" resourceid="england"
        href="DH?_MYMW_LANG=eng_US">English</mymw:link>
    </mymw:p>

  </mymw:body>
</mymw:document>
```

8.2 Annex 2 - Exemple d'una presentació amb tags JSTL

OPs/Telefons/generic/telefonos.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="firatarrega.css" type="text/css"?>
<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb"
xmlns:c="http://java.sun.com/jsp/jstl/core" id="telefonos">
  <mymw:head>
<mymw:title>Telèfons d'interès</mymw:title>
  </mymw:head>

  <mymw:body>

    <mymw:p id="p1" style="paginate">
      <c:forEach varStatus="status" begin="0" end="${index}" step="1">

        <c:choose>
          <c:when test="${_MYMW_LANG == 'es_ES'}">
<mymw:label style="labelbold">${telefonos.${status}.nom_es_ES}</mymw:label>
          </c:when>

          <c:when test="${_MYMW_LANG == 'eng_US'}">
<mymw:label style="labelbold">${telefonos.${status}.nom_eng_US}</mymw:label>
          </c:when>

          <c:otherwise>
<mymw:label style="labelbold">${telefonos.${status}.nom_ca_ES}</mymw:label>
          </c:otherwise>
        </c:choose>

        <mymw:label>${telefonos.${status}.telefon}</mymw:label>
        <mymw:label>${telefonos.${status}.descripcio}</mymw:label>

      </c:forEach>
    </mymw:p>

    <mymw:hr/>

    <mymw:include content="Common/generic/template/index"/>

  </mymw:body>
</mymw:document>
```

8.3 Annex 3 - Fitxer de configuració MyMobileWeb.Global.xml

config/MyMobileWeb.Global.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration name="root">
  <property name="LOG4J_Config" value="logs/traces.xml"/>
  <property name="Working_Mode" value="DEVL"/>
  <module name="Global">
    <property name="First_Application_OP" value="InitOp"/>
    <property name="Default_Language" value="ca_ES"/>
    <property name="Installed_Languages" value="eng_US,es_ES,ca_ES"/>
  </module>
  <module name="LoginManager">
    <property name="Login_Manager" value="org.morfeo.tidmobile.server.login.NoLogin"/>
  </module>
  <module name="FlowEngine">
    <property name="Flow_Handler" value="org.morfeo.tidmobile.server.flow.
      TidMobileFlowEngine"/>
    <property name="Handler_Package_Base" value="cat.iws.mobi.event"/>
  </module>
  <module name="PresentationAdapter">
    <property name="Presentations_Directory" value="jsp"/>
  </module>
  <module name="OAController">
    <property name="OA_Descriptor_File" value="OAConfig.xml"/>
  </module>
  <module name="URLManager">
    <property name="Context_Name" value="FiraTarrega"/>
    <property name="Navigate_Path" value="DH"/>
  </module>
  <module name="MessageManager">
    <property name="Messages_Directory" value="i18N/messages"/>
  </module>
  <module name="LiteralManager">
    <property name="Literals_Directory" value="i18N/literals"/>
    <property name="Global_Literals_Subdirectory" value="All"/>
  </module>
  <module name="ContentAdapter">
    <property name="Images_Path" value="resource/images"/>
  </module>
  <module name="Validator">
    <property name="Default_Date_Mask" value="dd/MM/yyyy"/>
    <property name="Default_Time_Mask" value="hh:mm:ss"/>
  </module>
  <module name="Transcoders">
    <module name="ImageTranscoder">
      <property name="Plugin_Class" value="org.morfeo.tidmobile.transcoder.TidImgTransc
    </module>
  </module>
</configuration>
```

8.4 Annex 4 - Exemple de full d'estil CSS

css/generic/firatarrega.css:

```
image.logo{
    transcode:true;
    align:center;
    weight-height:40;
    weight-width:40;
}

p.paginate{
    paginate: true;
    layout: vertical;
}

td.bold{
    color:black;
    font-weight:bold;
}

menu.mymenu{
    layout:vertical;
    align:left;
    colored:true;
    background-color1:#fff2b4;
    background-color2:#FFFFFF;
    white-space:nowrap;
    width:90%;
    ordered-list:true;
}

link.black{
    color:black;
}

link.blackbold{
    color:black;
    font-weight:bold;
}

link.red{
    color:#9e0505;
}

p.mlang{
    layout: horizontal;
    align: center;
    paginate: false;
```

```
}

p.header{
    align: center;
}

p.peu{
    layout: vertical;
    align: center;
}

label.labelred{
    color: #9e0505;
}

label.labelblue{
    color: blue;
}

label.labelbold{
    font-weight:bold;
}

telephonecaller.blue {
    color:blue;
    img-display: both;
}

chainedmenu.chmenu {
    autoselect: all;
    paginate: true;
    display-as: list-of-links;
    color: black;
}

link.img{
    img-display:only;
    border-width:0;
}

tr.header {
    background-color:Navy;
    color:Yellow;
    font-weight:bold;
    font-index: -2;
    align: center;
}

tr.body {
    font-family:helvetica;
```

```
background-color:#FFFFC8;  
font-index: -2;  
color: black;  
}
```

```
table.mytable {  
border-width:1;  
cellpadding:3;  
cellspacing:2;  
border-color:#666666;  
paginate: true;  
sel-column:0;  
color:black;  
}
```

8.5 Annex 5 - Exemple d'una OA

cat.iws.mobi.oa.GetTelefons.java:

```
package cat.iws.mobi.oa;

import java.util.ArrayList;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.morfeo.tidmobile.server.oa.BasicApplicationOperation;
import org.morfeo.tidmobile.server.oa.OAException;
import org.morfeo.tidmobile.context.Context;

public class GetTelefons extends BasicApplicationOperation {

    public void execute(Context the_context) throws OAException {

        Configuration conf = new Configuration();

        conf.configure();

        SessionFactory sessionFactory = conf.buildSessionFactory();

        Session session = sessionFactory.openSession();

        String qry = "from cat.iws.mobi.bean.Telefon ORDER BY id";

        ArrayList telefons = (ArrayList) session.createQuery(qry).list();

        session.close();

        the_context.setElement("index", telefons.size()-1 );

        the_context.setElement("telefons", telefons);

    }

}
```

Bibliografia

- [1] Enter & Idate. "Mobile 2007: Mobile Markets & Trends - Facts & Figures". 2007.
- [2] netsize. "The netsize guide - Convergence: Everything's going mobile". 2007.
- [3] dotMobi. "Mobile Web Developer's Guide". 2007.
- [4] arsys. "Guía dominios .mobi para Desarrolladores". 2007.
- [5] Forum Nokia. "Nokia Web Browser Design Guide". 2007.
- [6] Java Hispano. "Manual Hibernate". 2003.
- [7] Telefónica I+D. "MyMobileWeb: Manual de usuario del componente de gestión de dispositivos". 2006.
- [8] Telefónica I+D. "MyMobileWeb: Guía Rápida de desarrollo de aplicaciones". 2006.
- [9] Telefónica I+D. "MyMobileWeb: Guía de referencia del lenguaje de definición de presentaciones". 2006.
- [10] Telefónica I+D. "MyMobileWeb: Manual del programador para cliente ligero". 2006.
- [11] Mobile Development Community disponible a <http://dev.mobi>
- [12] Morfeo Project disponible a <http://www.morfeo-project.com>